



TAMPEREEN TEKNILLINEN YLIOPISTO

TONI FADJUKOFF

KARTTARAJOITTEIDEN KÄYTTÄMINEN
SISÄTILAPAIKANNUKSESSA

Diplomityö

Tarkastajat: Prof. Robert Piché ja
TkT Simo Ali-Löytty
Tarkastajat ja aihe hyväksytty
Luonnontieteiden ja ympäristötekniikan
tiedekunnan kokouksessa 4.4.2012

TIIVISTELMÄ

TAMPEREEN TEKNILLINEN YLIOPISTO

Teknis-luonnontieteellinen koulutusohjelma

FADJUKOFF, TONI: Karttarajoitteiden käyttäminen sisätilapaikannuksessa

Diplomityö, 57 sivua, 3 liitesivua

Helmikuu 2013

Pääaine: Matematiikka

Tarkastajat: TkT Simo Ali-Löytty ja prof. Robert Piché

Avainsanat: sisätilapaikannus, partikkelisuodatin, Bayesin menetelmät, karttarajoitteet, reaaliaikainen paikannus

Tässä diplomityössä käsitellään sisätilapaikannuksen ongelmaa ja esitetään siihen eräs ratkaisu käyttäen partikkelisuodatinta. Työssä esitellään sisätilapaikannukseen käytettyjä menetelmiä, joista erityisesti tarkastellaan inertiaalimittalaitteiden avulla suoritettavaa askeltunnistusta, sekä WLAN-mittausten perusteella paikannusta sisätiloissa.

Työssä esitellään bayesiläisen suodatuksen teoriaa pohjaksi partikkelisuodattimelle. Monte Carlo -laskentaan perustuvan partikkelisuodattimen algoritmi sovitetaan sisätilapaikannuksen malleihin.

Partikkelisuodattimen käyttämistä karttojen kanssa testataan ensin simulaation avulla. Työ esittelee muutaman menetelmän kartan hyödyntämiseksi suodattimessa, ja näiden hyödyllisyyttä arvioidaan simuloitujen testien avulla.

Osana työtä toteutettava reaaliaikainen paikannusjärjestelmä on toteutettu MATLAB-ympäristöön. Työssä esiteltyjä algoritmeja testataan Tampereen teknillisen yliopiston tiloissa. Suodatin hyödyntää rakennuksen tiloista kerättyjä WLAN-tietokantoja ja pohjapiirroksista muodostettuja karttoja ja saavuttaa noin 3 metrin mediaanivirheen reaaliaikaisena.

Työssä mietitään myös sisätilapaikannusta monessa kerroksessa. Mallia laajennetaan tukemaan useampaa kerrosta. Barometriä käytetään tunnistamaan kerroksen vaihtuminen.

ABSTRACT

TAMPERE UNIVERSITY OF TECHNOLOGY

Master's Degree Programme in Science and Engineering

FADJUKOFF, TONI: Using Map Constraints in Indoor Positioning

Master of Science Thesis, 57 pages, 3 Appendix pages

February 2013

Major: Mathematics

Examiners: Dr.Tech Simo Ali-Löytty and Prof. Robert Piché

Keywords: indoor positioning, particle filter, Bayes methods, map constraints, real time localisation

In this thesis work the indoor positioning problem is considered and one solution with the particle filter method is presented. Methods previously used for indoor positioning especially methods based on inertial measurements which are the base of pedestrian dead reckoning (PDR) -algorithms and Wi-Fi indoor positioning methods are researched.

The work presents the Bayesian filtering framework which is the basis of the particle filtering methods. A model which incorporates indoor maps is used with the sequential Monte Carlo based algorithms.

A simulator is programmed to test different map filtering techniques with the particle filter. The work presents several methods for using maps and their effects on the accuracy of the estimate are tested with simulated data.

A real time positioning software running in MATLAB is programmed as a part of this thesis work. The algorithms presented in this work are tested at the Tampere University of Technology. The filter uses gathered Wi-Fi data and the indoor maps generated from the floormap to achieve a median accuracy of 3 meters in real time.

Indoor positioning in multiple floors is considered. The model is expanded to support several floors. Barometer measurements are used to detect floor changes.

ALKUSANAT

Tämä työ on tehty Tampereen teknillisen yliopiston henkilökohtaisen paikannuksen algoritmien tutkimusryhmässä työskentelyn ohella. Aiheen valintaa ja työn tekoa on tukenut Nokian rahoittama tutkimus.

Haluan kiittää työni ohjaajia, professori Robert Pichéä ja TkT Simo Ali-Löyttyä kommenteista ja ideoinnista työn eri vaiheissa. Kiitän myös työkavereitani Henri Nurmis-
ta, Anssi Ristimäkeä ja Matti Raitoharjua työhön liittyvistä keskusteluista. Kiitokset
kuuluvat myös Miila Martikaiselle ja Shweta Shresthalle, jotka ovat auttaneet tässä
työssä käytettyjen mittausten keräämisessä.

Kiitän opiskelutovereitani opiskelujeni aikana kaikista koulussa ja sen ulkopuolella ko-
kemistamme yhteisistä hetkistä. Lopuksi haluan kiittää kotona häärivää tyttölaumaani
Mannaa, Emmiä ja Tiliä iltojeni piristyksestä ja vaimoani Lauraa saamastani tuesta
opiskelujen ja diplomityön kirjoittamisen aikana.

Tampere, 14. tammikuuta 2013

Toni Fadjukoff
Näyttelijäkatu 19 F 25
33720 Tampere

SISÄLTÖ

1. Johdanto	1
2. Sisätilapaikannuksen menetelmiä	3
2.1 Jalankulkijan arviointi	4
2.2 WLAN-pohjainen paikannus	6
2.2.1 Kuuluvuusaluepaikannus	7
2.2.2 Deterministinen paikannus radiokartalla	8
2.2.3 Probalistinen paikannus radiokartalla	9
2.2.4 Signaalin vaimentumiseen perustuvat mallit	9
2.3 Karttojen käyttäminen paikannuksessa	11
3. Matemaattinen tausta	13
3.1 Bayesiläinen suodatus	13
3.2 Kalmanin suodatin	14
3.3 Partikkelisuodatin	15
3.4 Työssä käytettävät mallit	20
4. Reaaliaikajärjestelmän toteuttaminen	23
4.1 Laitteisto	23
4.2 Karttainformaatio	24
4.2.1 Partikkelien ja seinien törmäystarkastelu	25
4.2.2 Seinien välttäminen	27
4.2.3 Seinään törmäämisen ennustaminen	28
4.2.4 Ihmisten käyttämien reittien suosiminen	31
4.2.5 Esiteltyjen painotusmenetelmien vertaaminen	31
4.2.6 Kartan virheet	32
4.3 Toteutus	32
4.3.1 Reaaliaikaiset mittaukset	34
4.3.2 Partikkelien mallinnus	36
4.3.3 Tarvittava partikkelien määrä	38
4.3.4 Ohjelman optimointi	39
4.3.5 Ohjelman käyttöliittymä ja rakenne	41
4.4 Testaustulokset	41
4.4.1 Testidatan mittaaminen	41
4.4.2 Tulokset	43
4.4.3 Tulosten analysointia	46
5. Monen kerroksen ongelma	48
5.1 Mallin muutokset	48
5.2 Barometrin mittauksen käyttäminen	49
5.3 Testireitti	50
6. Johtopäätökset	53
Lähdeluettelo	54
A. Paikannustuloksia	58

MERKINNÄT

$\int f(x) \, dx$	funktion f integraali
$f : A \rightarrow B$	funktio joukosta A joukkoon B
\propto	verrannollinen, myös todennäköisyysjakaumista
\approx	lähes sama
$\sum_{i \in I} a_i$	summa, jonka indeksit osoittaa joukko I
$\{a_i\}_{i=1}^n$	alkiot $a_1 \dots a_n$ sisältävä joukko
$(a_i)_{i=1}^n$	alkiot $a_1 \dots a_n$ sisältävä jono
$a \in A$	joukko A sisältää alkion a
A	matriisi
A^{-1}	matriisin A käänteismatriisi
$A_{(i)}$	matriisin A i :s vaakarivi
$ A $	matriisin A determinantti
$C(\mathbf{x}, d)$	yhteyspredikaatti, tosi, jos tilan \mathbf{x} paikasta on yhteys d kerroksen päähän
d_z	kerroksen muutos, diskreetti arvo
$N(\mu, \Sigma)$	normaalijakauma, jonka odotusarvo on μ ja kovarianssi Σ
$n_{\text{kalibrointi}}$	radiokartassa kalibrointipisteiden määrä
n_{mittaus}	mittauksessa kuultujen tukiasemien määrä
N_{eff}	efektiivinen näytteiden määrä
$p(\mathbf{x} \mathbf{y})$	\mathbf{x} :n todennäköisyys ehdolla \mathbf{y}
\mathbf{x}_k^i	partikkelin i tila ajanhetkellä k
\mathbb{R}^n	n -ulotteiset reaalityöt
$U(a, b)$	tasajakauma parametrein a ja b
w_k^i	partikkelin i paino ajanhetkellä k

\tilde{w}_k^i	partikkelin i normalisoimaton paino ajanhetkellä k
\mathbf{x}	vektori
$\hat{\mathbf{x}}$	vektorin \mathbf{x} suuntainen yksikkövektori
$\overrightarrow{\mathbf{x}, \mathbf{y}}$	pisteitä \mathbf{x} ja \mathbf{y} yhdistävä jana
$\ \mathbf{x}\ $	vektorin \mathbf{x} euklidinen normi
$\mathbf{x} \cdot \mathbf{y}$	vektorien pistetulo
$x \sim \mathbf{N}$	x noudattaa jakaumaa \mathbf{N}

LYHENTEET

BSSID	langattomassa lähiverkossa lähetettävä tukiaseman tunnus (engl.: Base Service Set Identifier)
CA	kuuluvuusalue (engl.: Coverage Area)
CUDA	Compute Unified Device Architecture
ESS	efektiivinen näytteiden määrä (engl.: Effective Sample Size)
ENU	paikallinen itä, pohjois, ylös -koordinaatit (engl.: local East North Up coordinates)
FP	sormenjälki (engl.: Fingerprint)
GPS	Global Positioning System
IMU	inertiaalimittalaite (engl.: Inertial Measurement Unit)
JIT	Just-in-time, ohjelman kääntäminen konekieleksi ajon aikana
KF	Kalmanin suodatin (engl.: Kalman Filter)
KNN	K:n lähimmän naapurin luokitusmenetelmä (engl.: K Nearest Neighbour)
LLH	Leveyspiiri, pituuspiiri ja korkeus -koordinaatisto (engl.: Latitude, Longitude and Height)
MAC	MAC-osoite, verkkosovittimen uniikki tunniste (engl.: Media Control Access)
NFC	Near Field Communication
NN	lähimmän naapurin luokitusmenetelmä (engl.: Nearest Neighbour)
PDR	Pedestrian Dead Reckoning
PF	partikkelisuodatin (engl.: Particle Filter)
PL	signaalin heikkeneminen, Path Loss
RAM	keskusmuisti (engl.: Random Access Memory)
RFID	Radio Frequency Identification
RSS	Vastaanotetun signaalin voimakkuus (engl.: Received Signal Strength)

SLAM	Simultaneous Localization And Mapping)
TTY	Tampereen teknillinen yliopisto
UWB	Ultra Wideband
WLAN	langaton lähiverkko (engl.: Wireless Local Area Network)
WKNN	painotettu K:n lähimmän naapurin luokitusmenetelmä (engl.: K Nearest Neighbours)
WSN	langaton sensoriverkko (engl.: Wireless Sensor Network)
ZUPT	Zero Velocity Update

1. JOHDANTO

Tässä työssä tutkitaan sisätilapaikannusta ja sen ongelmia. Maailmanlaajuisesti paikannukseen käytetään GPS (Global Positioning System) -järjestelmää. Monet älypuhelimet ja muut kulutuslaitteet sisältävät GPS-vastaanottimen ja sen ympärille rakennetun kartta- ja navigointiohjelmiston. Karttaa käytetään hyödyksi GPS-sovelluksissa sekä lasketun estimaatin korjaamiseen kartalle sopivaan paikkaan että käyttäjälle tilanteen visualisoimiseen. GPS-pohjainen paikannus on tarkkaa, mutta GPS:n toimintavarmuus sisätiloissa on vähintäänkin epävarmaa, sillä rakennusten rakenteet häiritsevät satelliittien lähettämiä GPS-signaaleja liikaa. Sisätiloissa on siis käytettävä muita keinoja paikannukseen.

Sisätiloissa paljon paikannukseen käytettyjä mittauksia ovat havainnot WLAN-tukiasemista ja erilaisista langattomien sensoriverkkojen sensoreista. Langattomista verkoista saatuja mittauksia voidaan käyttää hyväksi henkilön paikannukseen monin tavoin. Vastaanotetun signaalin voimakkuudelle (received signal strength, RSS) on olemassa malleja, joissa signaalin voimakkuutta voidaan arvioida etäisyyden funktiona, kunhan ympäristölle ja radiolaitteille ominaiset parametrit tunnetaan. Myös sormenjälkiin (fingerprinting, FP) perustuvat menetelmät ovat tutkittuja. Menetelmissä verrataan saatuja WLAN-mittauksia aiemmin mitattuun dataan ja yritetään löytää mahdollisimman samanlaiset mittaukset.

Signaalien lisäksi myös inertiamittalaitteita (inertial measurement units, IMU) voidaan käyttää henkilön paikantamiseen sisätiloissa. Tämä menetelmä tunnetaan nimellä jalankulkijan paikan arviointi (pedestrian dead reckoning, PDR). IMU-sensorit voivat olla erityisissä sensorilaitteissa, tai ne voidaan sisällyttää muuhun elektroniikkaan kuten kannettaviin tietokoneisiin tai matkapuhelimiin.

Ihmiselle luonnollinen tapa suunnistaa sisätiloissa on katsoa kartasta nykyinen sijainti ja kohde sekä hahmottaa näiden perusteella reitti kohteeseen. Monissa kauppakeskukissä ja vastaavissa on olemassa sisätilakarttoja, joissa eri kohteiden nimet sekä nykyinen paikka on ilmoitettu. Tässä työssä keskitytään erityisesti karttojen käyttämiseen apuna sisätilapaikannuksessa.

Tämän työn toisessa luvussa esitellään sisätilapaikannuksen eri menetelmiä. Erityinen huomio kiinnitetään WLAN-mittauksiin pohjautuviin menetelmiin, joita tässäkin työssä tutkitaan tarkemmin. Myös sensoripohjaiset paikannusmenetelmät esitellään kattavasti.

Työn kolmannessa luvussa käydään paikannusohjelman kannalta tärkeät matemaattiset mallit läpi. Vaihtoehtoisia malleja pohditaan lyhyesti. Suodatuksen idea esitellään Kalmanin suodattimen avulla. Sisätilapaikannuksen kannalta tärkeän partikkelisuodattimen matemaattisia periaatteita ja algoritmeja esitellään.

Työn osana toteutetaan sisätilapaikannusohjelma. Ohjelma käyttää hyväkseen XSens-sensorilaitetta, WLAN-mittauksia ja sisätilakarttoja. Neljänteen lukuun dokumentoidaan järjestelmän toteutus ja algoritmit. Osana reaaliaikajärjestelmää toteutetaan myös malleille simulaattori, jonka avulla eri ratkaisuja voidaan testata nopeammin ja tarkemmin kontrolloidussa ympäristössä. Neljännessä luvussa pohditaan myös monia käytännön ongelmia, kuten virheellisiä karttoja ja partikkelisuodattimen ajautumista karille.

Työn viidennessä luvussa käsitellään vielä partikkelisuodattimen käyttämistä kartan eri kerroksissa. Mallit laajennetaan tukemaan useaa kerrosta. Reaaliaikaista monen kerroksen toiminnallisuutta havainnollistetaan todellisiin mittauksiin perustuvalla esimerkillä. Useassa kerroksessa toimivan paikannusjärjestelmän jatkokehitystä pohditaan.

2. SISÄTILAPAUKANNUKSEN MENETELMIÄ

Tässä luvussa esitetään erilaisia sisäpaikannusratkaisuja. Aluksi tehdään hyvin pikainen katsaus käytettyihin menetelmiin, jonka jälkeen esitellään tässä työssä käytettäviä menetelmiä tarkemmin.

Sisätilapaikannusta on tutkittu viimeisen vuosikymmenen aikana hyvin laajasti. Päinvastoin kuin ulkotiloissa, joissa toimivat maailmanlaajuiset satelliittipaikannusjärjestelmät kuten GPS tai GLONASS, sisätilapaikannukseen ei ole vielä olemassa standardimenetelmää. Monia eri teknologioihin perustuvia menetelmiä on tutkittu ja joitain järjestelmiä on lanseerattu kaupalliseen käyttöön.

Useat näistä järjestelmistä perustuvat langattomien signaalien vastaanottamiseen ja paikantamiseen saapuvan signaalinvoimakkuuden perusteella (received signal strength). UWB (Ultra Wideband) -signaalien käyttöä on tutkittu signaalin vaimenemiseen perustuvan mallin kanssa. Menetelmällä saadaan paikannustarkkuudeksi 1,5 metriä [11]. RFID (Radio Frequency Identification) -menetelmä perustuu radiotaajuuksien etätunnistukseen. Näitä signaaleja on käytetty mm. sormenjälkiin perustuvassa probabilistisessä menetelmässä. Järjestelmän virhe on muutama metri [4]. RFID-teknologiaan perustuvaa hyvin lyhyen matkan NFC (Near Field Communication) -tekniikkaan perustuvia sisätilapaikannusratkaisuja on myös tutkittu. NFC-tekniikan avulla on luotu interaktiivisia paikannusjärjestelmiä, joissa käyttäjä pystyy lataamaan sisätilakartan ja määränpäänsä sijainnin omaan NFC-tuettuun päätelaitteeseensa sisätiloihin asennettujen NFC-tagien luona. Käyttäjä voi myös päivittää sijaintinsa päätelaitteeseensa näiden NFC-tagien luona [28]. Langaton sensoriverkko (engl.: Wireless Sensor Network) on paikallinen verkko, joka on rakennettu edullisista sulautetuista mikrokontrollereista, joissa on ainakin radiolähetin ja -vastaanotin. Esimerkiksi tutki-

muksessa [20] käytetään langatonta sensoriverkkoa kuuluvuusaluepaikannuksessa noin 10 metrin keskivirheellä.

Bluetooth kehitettiin alunperin tiedonsiirtoon, mutta tekniikkaa on jo pidempään tutkittu sisätilapaikannuksessa. Bluetooth on useissa mobiileissa laitteissa sisäänrakennettuna, ja sitä käyttäviä järjestelmiä on suhteellisen halpa tuottaa. Myös Bluetooth-tekniikka perustuu vastaanotettuihin signaaleihin ja lähettimistä rakennettuun tietokantaan. [6; 26]

Pseudoliittien eli pseudo-satelliittien varaan rakennettua sisätilapaikannusta on myös tutkittu. Pseudoliitti on laite, joka matkii toiminnaltaan ja signaaleiltaan GPS-järjestelmän satelliitteja. Se ei kuitenkaan ole oikea satelliitti, vaan sijaitsee maan päällä. Pseudoliittien toiminta on lähtökohtaisesti monimutkaisempaa kuin edellä esitettyihin teknologioihin perustuvilla menetelmillä ja muistuttaa enemmän GPS-järjestelmän toimintaa. Esimerkiksi laitteiden kellojen on oltava synkronoitu, jotta paikannusjärjestelmä voi toimia. [2]

Kameralla otettuihin kuviin perustuvaa paikannusta on myös tutkittu. Esimerkiksi tunnistamalla kuvasta reunoja ja niiden avulla katoamispisteet, voidaan rakentaa kameran avulla gyroskooppi. Menetelmä toimii hyvin sisätiloissa, joissa ihminen on rakentanut lähes kaikki viivat ortogonaalisesti. [16] Kameraa voidaan käyttää myös tunnistettavien kohteiden kuvaamiseen. Kuvasta voidaan tunnistaa kohteet ja näiden avulla paikantaa sekä kameran sijainti että orientaatio. Tunnisteina voi olla esimerkiksi kuvassa näkyvät ovet, jotka voidaan sovittaa tietokannassa löytyviin ovien sijainteihin. [34]

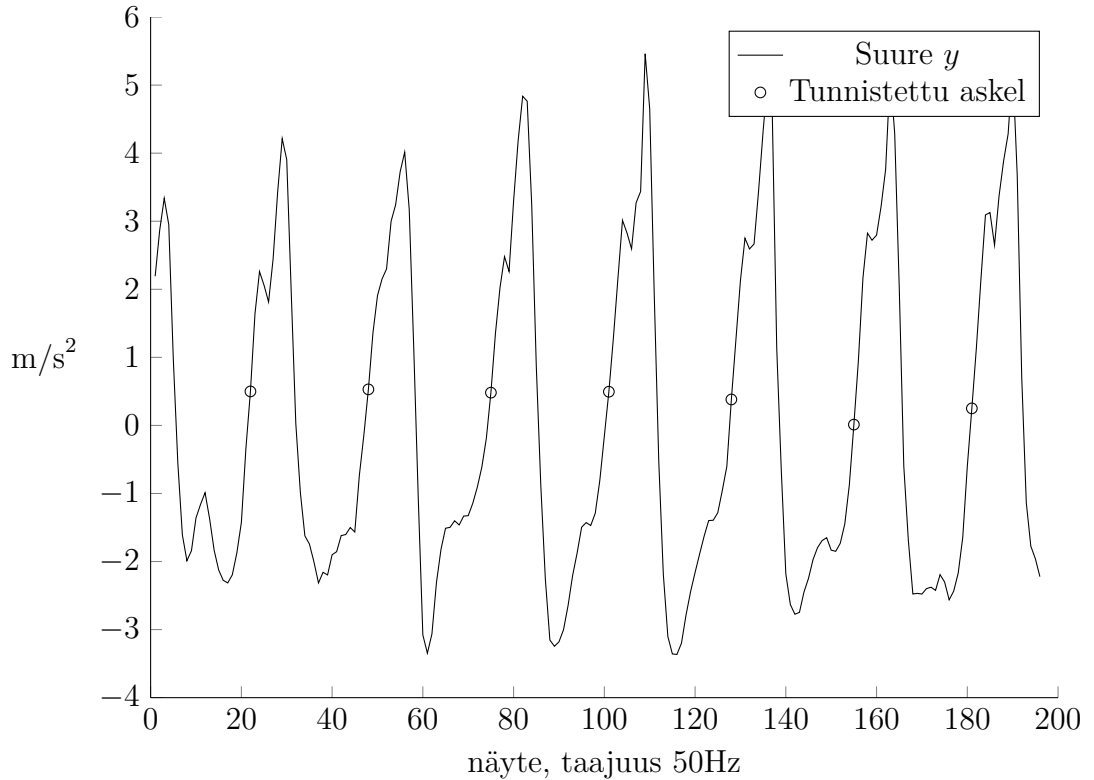
Uusimpia sisätilapaikannusteknologioita edustaa maan magneettikenttään perustuva paikannus. Samoin kuin signaaleihin perustuvissa menetelmissä, on ensin rakennuksesta luotava magneettikenttäkartta. Indoor Atlaksen teknologia saavuttaa jopa alle kahden metrin tarkkuuden. [22]

2.1 Jalankulkijan arviointi

Jalankulkijan arviointi (engl.: Pedestrian Dead Reckoning) on menetelmä, jossa sensorihavaintojen avulla pyritään mallintamaan ihmisen kävelyä. Jalankulkijan vaatteisiin liitetään sensoreita kuten kiihtyvyysanturi, magnetometri, barometri eli ilmapuntari ja gyroskooppi. Dead Reckoning -menetelmät pyrkivät mallintamaan sijainnin muutosta. Menetelmän periaatteena on arvioida sijaintia kuljetun matkan ja suunnan perusteella. Toiminta perustuu kahteen vaiheeseen: askelten tunnistamiseen ja suunnan arvioimiseen.

Askeltunnistus tehdään yleensä kiihtyvyysantureiden avulla. Kiihtyvyyden suunnasta voi usean mittauksen aikavälillä arvioida painovoiman suuntaan sensoriin nähden.

Käveleminen ja askeleet näkyvät mitatussa datasta jaksoina. Jaksot voidaan tunnistaa esimerkiksi vähentämällä kiihtyvyydestä painovoiman aiheuttama kiihtyvyys ja tarkastella sen jälkeen kiihtyvyyden nollan ylitystä (zero crossing). Tunnistuksen toiminnan takaamiseksi usein asetetaan raja-arvot, jotka kiihtyvyyden täytyy ylittää, jotta askel tunnistetaan. Tällä pyritään estämään askeleen virheellistä tunnistamista mittauslaitteen epätarkkuuden tai kädessä pitelyn aiheuttaman pienen kiihtyvyyden takia. Kuvassa 2.1 on piirretty kiihtyvyydestä johdettu arvo $y = \|\mathbf{a}\| - \|\mathbf{g}\|$ ajan suhteen kävelyn aikana. Kiihtyvyyden normista $\|\mathbf{a}\| = \sqrt{a_x^2 + a_y^2 + a_z^2}$ vähennetään painovoiman aiheuttama kiihtyvyys $\|\mathbf{g}\| = 9,81\text{m/s}^2$. Askel tunnistetaan kun suureen y merkki vaihtuu negatiivisesta positiiviseksi. Askel-tunnistusta on mahdollista parantaa usein keinoilla. Kiihtyvyyden normi voidaan alipäästösuodattaa, esimerkiksi laskemalla liukuvaa keskiarvoa, jolloin mittauksessa esiintyvä korkeiden taajuuksien kohina pienenee. Voidaan myös vaatia askeleiden välillä tietty aika ennen seuraavan askeleen hyväksymistä tai y -arvoa ylittämään tai alittamaan asetettu kynnyksen ennen askeleen hyväksymistä. [3] Kuvan askeltunnistuksessa ei ole käytetty kynnyksiarvoja.



Kuva 2.1: Kiihtyvyysanturista mitattua kiihtyvyyttä. Kuvaa on piirretty kiihtyvyyden normi. Tunnistettut askeleet on merkitty mustalla pallolla.

Magnetometriä voidaan käyttää kompassina, ja näin ollen voidaan saada selville laitteen asento maapallon magneettiseen napaan nähden. On huomattava, että monet

metallirakennelmat vaikuttavat magnetometrin mittauksiin. Näin ollen monissa rakennuksissa magneettikenttään syntyvät häiriöt laskevat magnetometriin perustuvan kompassin tarkkuutta. Gyroskoopin mittauksina saadaan kulmanmuutoksen nopeutta. Gyroskooppi antaa yleensä tarkkoja mittauksia, mutta toisaalta sen avulla voidaan mitata vain suhteellista kääntymistä. Gyroskoopin virhe ei useinkaan ole valkoista kohinaa, vaan mitattu suunta systemaattisesti luisuu kauemmas alkuperäisestä suunnasta.

Gyroskoopin, magnetometrin ja kiihtyvyysanturin avulla voidaan rakentaa järjestelmä, joka pystyy laskemaan asentonsa. Tällainen monen sensorin algoritmi on käytössä esimerkiksi XSens-laitteessa [38]. Ilman gyroskooppia on kuitenkin mahdollista muodostaa hyvä estimaatti suunnasta [8].

Korkeutta esimerkiksi merenpinnasta arvioidaan korkeusmittarin mittaamasta ilmanpaineesta. Korkeustietojen saaminen ei varsinaisesti auta jalankulkijan liikkeen selvittämiseen, mutta sen avulla voidaan arvioida jalankulkijan sijaintia kartalla.

Jos kiihtyvyysanturi on liitetty ihmisen jalkaan, ja kaikkien sensorien tarkkuus on tarkka, voidaan sijainnin muutoksia arvioida integroimalla sensorien antamaa kiihtyvyyttä. Jotta askeleen pituus saadaan selville, on selvitettävä ensin hetket, jolloin jalka on maassa. Tätä kutsutaan nimellä englanniksi nimellä Zero Velocity Update (ZUPT). Kahden ZUPT-tilan välillä jalka ottaa yhden askeleen, verrattuna kahteen askeleeseen. Tätä väliä kutsutaan harppaukseksi. Paikan muutosta voidaan arvioida integroimalla harppauksen aikana sensoreista mitattua kiihtyvyysdataa. Tällaisella menetelmällä voidaan päästä jopa 1 – 2% virheeseen todellisesta matkasta. [13]

Dead Reckoning -menetelmä on sopiva vain paikan päivitykseen. Menetelmällä voidaan laskea vain paikan muutosta, eikä sillä voida paikantaa ilman alkupaikkaa ja suuntaa. Useat menetelmää käyttävät paperit olettavatkin alkusijainnin tunnetuksi.

Alkupaikan arviointiin voidaan käyttää esimerkiksi GPS-mittausta, jos sellainen on saatavilla. Muita vaihtoehtoja ovat käyttäjän antama syöte, tai paikallisesti kerätyn datan perusteella tehty estimaatti. Tällainen voi olla esimerkiksi Bluetooth tai WLAN-pohjainen paikannus.

Koska uutta sijaintia estimoidaan aina edellisen sijainnin perusteella, ilman toistuvia ulkoisia mittauksia menetelmän virhe kasvaa rajatta. [24]

2.2 WLAN-pohjainen paikannus

Rakennuksissa olevat WLAN-tukiasemat ovat yleistyneet todella paljon viimeisen vuosikymmenen aikana. Monet radioteknologiaan perustuvat paikannusjärjestelmät vaativat lisäinvestointeja ja laitteiston ostamista ja asentamista tätä tarkoitusta varten,

mutta WLAN-verkkoihin perustuvat järjestelmät on mahdollista toteuttaa ilman suuria lisäkustannuksia käyttäen jo olemassa olevaa infrastruktuuria. WLAN-asemien avulla paikantamiseen perustuvia järjestelmiä onkin tutkittu erityisen paljon viime aikoina. WLAN-mittausten avulla voidaan saavuttaa suhteellisen hyvä paikannustarkkuus itseopitullakin radiokartalla [30].

Sormenjälkimenetelmillä tarkoitetaan WLAN-paikannuksessa menetelmiä, joissa ensin kerätään sisätiloista sormenjälkiä. Nämä sormenjäljet sisältävät tiedon kuulluista WLAN-tukiasemista ja saapuneen signaalin voimakkuuden.

Sormenjäljistä muodostuu radiokartta. Radiokartta voidaan määrittellä joukkona sijainteja, joihin liitetään sormenjäljen sisältävä tieto. Kartan i :s elementti voidaan määrittellä

$$M_i = (B_i, \{\mathbf{a}_{ij} | j \in N_i\}, \boldsymbol{\theta}_i), \quad 1 \leq i \leq n_{\text{kalibrointi}} \quad (2.1)$$

missä B_i on solu, jonka keskus \mathbf{p}_i on i :s kalibrointipiste, jossa sormenjälki on mitattu. Sormenjäljen vektori \mathbf{a}_{ij} määrittää RSS-arvojen vektorin mittauksessa j . Käytetystä lähteestä riippuen $\boldsymbol{\theta}_i$ sisältää muuta tietoa, kuten tieto mittauslaitteen orientaatiosta. [15]

Radiokartan muodostamisen aikana siihen on mahdollista liittää paikannusvaiheessa käytettäviä arvoja. Esimerkiksi tunnuslukuja kuten kuultujen RSS-arvojen keskiarvo ja varianssi voidaan laskea eri tukiasemille erikseen ja liittää radiokarttaan. [15]

2.2.1 Kuuluvuusaluepaikannus

Kuuluvuusaluemallissa jokaiselle tukiasemalle määritetään kuuluvuusalue. Yksi tapa mallintaa kuuluvuusaluetta on normaalijakauma. Tällöin voidaan ajatella, että todennäköisyys kuulla jokin tukiasema paikassa \mathbf{z} noudattaa normaalijakaumaa $p(\mathbf{z} | \boldsymbol{\mu}, \Sigma) \sim N(\boldsymbol{\mu}, \Sigma)$. Parametrien $\boldsymbol{\mu}$ ja Σ estimoimiseksi voidaan rakentaa kaksiulotteinen mittausmalli

$$\mathbf{Y} = \mathbf{1}\boldsymbol{\mu}^T + \boldsymbol{\epsilon}, \quad (2.2)$$

missä \mathbf{Y} on $n_{\text{kalibrointi}} \times 2$ -kokoinen mittausmatriisi sisältäen kalibrointipisteet \mathbf{p}_i , joissa tukiasema on kuultu, $\boldsymbol{\mu}$ on sijaintia kuvaava 2-ulotteinen vektori ja $\boldsymbol{\epsilon}$ on $n_{\text{mittaus}} \times 2$ -kokoinen virhematriisi. Kun virheen vaakavektorit $\boldsymbol{\epsilon}_{(i)} = \mathbf{Y}_{(i)} - \boldsymbol{\mu}^T$ oletetaan riippumattomiksi ja gaussisesti jakautuneiksi, $\boldsymbol{\epsilon}_{(i)} | \boldsymbol{\mu}, \Sigma \sim N(\mathbf{0}, \Sigma)$, saadaan ratkaistua optimaalinen kuuluvuusalue bayesiläisittäin käyttäen odotusarvon priorina gaussista jakaumaa ja kovarianssin priorina käänteistä Wishartin jakaumaa. [19]

Paikannusvaiheessa, kun kuuluvuusalue tietokanta on muodostettu, saadaan paikan jakauma estimoitua yhtälöillä $\hat{x} \sim N(\mathbf{x}, \mathbf{C})$, missä

$$\mathbf{x} = \left(\sum_{i=1}^{n_{\text{mittaus}}} \Sigma_i^{-1} \right)^{-1} \left(\sum_{i=1}^{n_{\text{mittaus}}} \Sigma_i^{-1} \boldsymbol{\mu}_i \right) \quad (2.3)$$

ja

$$\mathbf{C} = \left(\sum_{i=1}^{n_{\text{mittaus}}} \Sigma_i^{-1} \right)^{-1}, \quad (2.4)$$

missä indeksi i käy läpi kaikki mittauksessa kuullut tukiasemat ja käyttää aiemmin estimoituja parametreja hyväkseen.

Kuuluvuusaluepaikannuksella tarkkuus on sisätiloissa noin 10 metriä. [19]

2.2.2 Deterministinen paikannus radiokartalla

Jos tilaa \mathbf{x} ei oleteta satunnaismuuttujaksi, niin sen estimaatti voidaan laskea kalibrointipisteiden \mathbf{p}_i painotettuna keskiarvona.

$$\mathbf{x} = \sum_{i=1}^{n_{\text{kalibrointi}}} \frac{w_i}{\sum_{j=1}^{n_{\text{kalibrointi}}} w_j} \mathbf{p}_i \quad (2.5)$$

Painot w_i riippuvat radiokartan kalibrointipisteessä \mathbf{p}_i tehdyistä mittauksista. Paino voidaan laskea esimerkiksi RSS-innovaation normin käänteislukuna:

$$w_i = \frac{1}{\|\mathbf{y} - \mathbf{a}_i\|}, \quad (2.6)$$

missä \mathbf{y} on mitattujen RSS-arvojen vektori, ja \mathbf{a}_i radiokartassa olevat keskiarvot RSS-mittauksille kalibrointipisteessä i , ja $\|\cdot\|$ on jokin normi. Normina voidaan käyttää esimerkiksi Manhattanin normia, Euklidista normia tai Mahalanobiksen normia. [15]

Lähimmän naapurin (NN) menetelmässä valitaan \mathbf{x} :n estimaatiksi on se \mathbf{p}_i , jolla w_i on suurin. Menetelmää voidaan laajentaa painotetuksi K:n lähimmän naapurin (WKNN) menetelmäksi. Tällöin K suurinta painoa otetaan mukaan yhtälöön (2.5), ja loput asetetaan nolliksi. Jos jäljelle jääneet painot asetetaan yhtä suuriksi, on kyseessä K:n lähimmän naapurin menetelmä (KNN). Yleensä WKNN- ja KNN-menetelmät saavat parempia tuloksia kuin NN-menetelmä, mutta tiheällä radiokartalla myös NN saa hyviä tuloksia. [15]

2.2.3 Probalistinen paikannus radiokartalla

Toinen näkökulma mittausten ja sormenjälkien yhteensovittamiseen on bayesiläinen. Tällöin oletetaan tila \mathbf{x} satunnaisvektoriksi ja halutaan laskea posteriorijakauma mitauksia \mathbf{y} käyttäen.

Bayesin kaava on

$$p(\mathbf{x}|\mathbf{y}) = \frac{p(\mathbf{y}|\mathbf{x})p(\mathbf{x})}{p(\mathbf{y})}, \quad (2.7)$$

missä $p(\mathbf{y}|\mathbf{x})$ on uskottavuus, $p(\mathbf{x})$ on paikan prior, ja $p(\mathbf{y})$ normalisointivakio. Radiokarttaa (2.1) käytettäessä eräs sopiva prior on tasajakautunut radiokartan solujen suhteen

$$p(\mathbf{x}) \propto \sum_{i=1}^{n_{\text{kalibrointi}}} \chi_{B_i}(\mathbf{x}), \quad (2.8)$$

missä

$$\chi_{B_i}(\mathbf{x}) = \begin{cases} 1, & \mathbf{x} \in B_i \\ 0, & \mathbf{x} \notin B_i \end{cases}. \quad (2.9)$$

Uskottavuusfunktiona voidaan käyttää

$$p(\mathbf{y}|\mathbf{x}) = \sum_{i=1}^{n_{\text{kalibrointi}}} p(\mathbf{y}|i)\chi_{B_i}(\mathbf{x}), \quad (2.10)$$

missä

$$p(\mathbf{y}|i) = \prod_{j=1}^{n_{\text{mittaus}}} p_{\mathbf{v}_{ij}}(y_j - \bar{a}_{ij}). \quad (2.11)$$

Käytännössä siis radiokartan muodostamisvaiheessa kerättyjä RSS-arvoja verrataan mitattuihin RSS-arvoihin bayesiläisen todennäköisyysfunktion muodossa. Niin kutsutut ydintiheysestimaatit toimivat hyvin $p_{\mathbf{v}_{ij}}(\cdot)$ -tiheysfunktiona [15].

2.2.4 Signaalin vaimentumiseen perustuvat mallit

Fysiikan lakien avulla määritellään signaalin vaimenemiselle erilaisia arvioivia malleja ja paikannusongelman kannalta eräs helpoiten lähestyttävistä on log-etäisyys-malli

$$P(d) = P(d_0) - 10n \log_{10} \left(\frac{d}{d_0} \right) + v, \quad (2.12)$$

missä $P(d)$ on RSS-arvo etäisyydellä d tukiasemasta, d_0 mallissa käytettävä verrokietäisyys, n signaalin vaimenemiskerroin ja v langattoman signaalin varjostumista kuvaavaa valkoista kohinaa. Usein käytetään verrokietäisyytenä yhtä metriä, jolloin $d_0 = 1$ ja

$$P(d) = A - 10n \log_{10}(d) + v \quad (2.13)$$

Tällöin A on tukiaseman lähetyksen RSS-arvo metrin etäisyydellä tukiasemasta.

Fysiikan lakien mukaan parametri n saa tyhjiössä arvon 2, mutta sisätiloissa rakenteiden nostavat n :n arvoa ja tällöin $n \geq 2$. Lisäksi monimutkaisten sisätilojen kanssa hyvin usein seinät ja muut esteet häiritsevät signaalin kulkua ja esimerkiksi artikkelissa [5] esitetään kaksiosainen malli, jossa n saa eri arvon riippuen signaalin voimakkuudesta. Tällä pyritään mallintamaan esteiden vaikutusta signaalin voimakkuuteen.

Ennen vaimenemismalleihin perustuvaa paikannusta on estimoitava mallit paikannuksessa käytettäville tukiasemille. Tämä tehdään käyttämällä mitattuja sormenjälkiä.

Parametrien estimointi voidaan nähdä neliösumman optimointitehtävänä.

$$\theta(A, n, \mathbf{m}) = \sum_{i=1}^{n_{\text{kalibrointi}}} \|P_i - (A - 10n \log_{10}(\|\mathbf{m} - \mathbf{p}_i\|))\|^2 \quad (2.14)$$

Yhtälössä (2.14) parametrit A ja n ovat kuin edellä ja m on tukiaseman paikka. Arvot $P_i, i \in (1, N)$ ovat tukiasemasta saadut RSS-mittaukset paikoissa \mathbf{p}_i . Koska optimointitehtävä on epälineaarinen, sen ratkaisemiseen on käytettävä erilaisia optimointimenetelmiä, kuten Gauss-Newtonin iterointia. [27]

Saaduista parametriestimaateista paikantaminen onnistuu bayesiläisessä mielessä. Mittausfunktio on epälineaarinen

$$\mathbf{y} = h(\mathbf{x}, \mathbf{a}) + \mathbf{v}, \quad (2.15)$$

missä \mathbf{y} on saadut mittaukset, \mathbf{x} tuntematon tila, \mathbf{a} estimoidut tukiasemien parametrit jakaumineen ja \mathbf{v} valkoista kohinaa. Mallista voidaan muodostaa posteriorijakauma

$$p(\mathbf{x}|\mathbf{y}) = \frac{\int p(\mathbf{y}|\mathbf{a}, \mathbf{x})p(\mathbf{a}|\mathbf{x})p(\mathbf{x}) d\mathbf{a}}{\iint p(\mathbf{y}|\mathbf{a}, \mathbf{x})p(\mathbf{a}|\mathbf{x})p(\mathbf{a}) d\mathbf{a} d\mathbf{x}} \quad (2.16)$$

Posteriorin laskeminen ja näin ollen paikantaminen onnistuu esimerkiksi muodostamalla suuri hila (engl.: grid), jonka pisteissä lasketaan RSS-arvojen paikkaansapitävyys mallin suhteen ja tätä kautta todennäköisyys. Kun hilan muodostamiseen käytetään Bayesin menetelmien mukaista prioria, voidaan se rajoittaa pienemmäksi ja tällöin tarvittava laskenta-aika pienenee.

Muita menetelmiä estimaatin laskemiseen ovat Gauss-Newton -iterointiin perustuvat menetelmät tai Monte Carlo -menetelmiin perustuvat Metropolis-Hastings -menetelmät. [27]

2.3 Karttojen käyttäminen paikannuksessa

Perimmiltään paikannuksessa on kyse paikannettavien sijoittamisesta kartalle. Jotkut algoritmit käyttävät karttoja apunaan estimaatin laskemiseen, toiset vain esittävät tuloksensa kartan avulla visualisoituna.

GPS:n kanssa käytetään usein karttaa. Autojen navigointilaitteissa on kokonaisen maiden tieverkostoja tallennettuna jopa kaistan tarkkuudella. Monet laitteet sijoittavat kuljettajan autoineen tielle, vaikka GPS-estimaatti olisikin hieman sivussa. Tällaisia algoritmeja kutsutaan usein kartalla sovitus (engl.: map matching) -algoritmeiksi [29]. Algoritmien tarkoituksena on löytää sopiva tie, jota pitkin kuluneuvo saattaa kulkea. Usein tieverkosto esitetään jonkinasteisena graafina. Kartalle sovitus tehdään useimmiten reaaliaikaisena, mutta myös jälkikäteen sovittaminen on mahdollista ja joissain tapauksissa hyödyllistä [23].

Tiet eivät ole kuitenkaan ainoa mielenkiintoinen asia kartoissa. Jalankulkijat, maastosuunnistajat ja hiihtäjät eivät välttämättä kulje ollenkaan teitä pitkin, eikä heidän liikettään voi arvioida tiekartan avulla juurikaan. Kuitenkin esimerkiksi jyrkän teet ja rakennukset voisivat luoda rajoitteita paikannukseen.

Sisätilakartat eivät ole vielä vakiinnuttaneet asemaansa kuluttajaelektroniikassa. Niiden sisältämä informaatio muistuttaa kuitenkin ulkotilojen rajoitteita. Karttatietoon voidaan sisällyttää paikat, joihin on mahdollista kulkea, sekä paikat, joihin ei voi mennä. Sisätilakarttoihin on sovellettu myös tieverkon ja GPS:n kaltaisia kartalle sovitus -algoritmeja [12]. Näissä on kuitenkin huomioitava, että jalankulkijan liikkuminen on huomattavasti vapaampaa autoilijoihin verrattuna.

Useissa tutkimuksissa on hyödynnetty partikkelisuodatinta ja sisätilakarttoja. Karttojen on havaittu parantavan partikkelisuodatuksen paikannustarkkuutta [21]. Osassa töissä sensorimittauslaite on kiinnitetty jalkaan, jolloin askeleen pituutta ja korkeuseroja voidaan arvioida tarkasti [35; 37]. Tutkimuksessa [18] hyödynnetään sisätiloihin paikannustarkoituksessa asennettuja lähettämiä, joiden avulla voidaan mitata saapuvan signaalin kulma. Kulman ja trigonometrinen funktioiden avulla voidaan selvittää etäisyys lähettimestä. Yhdistämällä mittauksen partikkelisuodattimeen ja karttaan voidaan saavuttaa hyvä estimaatti ilman alkusijaintia ja kädessä pidettävän sensorilaitteen avulla.

Itseoppivia järjestelmiä on tutkittu sisätilapaikannuksessa. SLAM (Simultaneous Localization And Mapping) -menetelmässä ihmisen selkään kiinnitettävä laite tekee mittauksia, joiden avulla voidaan estimoida sisätilakarttaa. Menetelmä yhdistää digitaalisen valokuvauksen ja inertiaalisensorit. [25]

Tässä työssä kuvattavissa sisätilakartoissa oletetaan huoneiden nurkkakoordinaatit tunnetuiksi. Lisäksi huoneet voidaan yhdistää toisiinsa puhkaisemalle seinään ovia tai sopimalla jokin tila hissiksi tai portaikoksi.

3. MATEMAATTINEN TAUSTA

Tässä luvussa käydään läpi partikkelisuodattimen matemaattista taustaa ja algoritmia. Myös Kalmanin suodatin esitellään.

3.1 Bayesiläinen suodatus

Bayesiläisessä suodatuksessa on tarkoituksena arvioida systeemin tilan todennäköisyysjakaumaa. Tilaa diskreetillä ajanhetkellä t_k merkitään \mathbf{x}_k . Ajanhetkellä t_k saatua mittausta merkitään \mathbf{y}_k .

Bayesiläisessä suodatuksessa oletetaan Markovin malli, jolloin tilan todennäköisyysjakauma riippuu vain edellisestä tilasta.

$$p(\mathbf{x}_k | \mathbf{x}_1, \dots, \mathbf{x}_{k-1}) = p(\mathbf{x}_k | \mathbf{x}_{k-1}) \quad (3.1)$$

Mittausten oletetaan myös riippuvan ainoastaan nykyisestä tilasta.

$$p(\mathbf{y}_k | \mathbf{x}_1, \dots, \mathbf{x}_k) = p(\mathbf{y}_k | \mathbf{x}_k) \quad (3.2)$$

Suodatuksen alkujakauma ei riipu mallin mittauksista, eikä muista tiloista.

$$p(\mathbf{x}_0) = p(\mathbf{x}_0 | \mathbf{y}_{1:0}) \quad (3.3)$$

Bayesiläinen suodatus on rekursiivista, eli jakauman estimaattia päivitetään saatujen mittausten avulla. Ehdollistamalla tila \mathbf{x}_k mittauksen $\mathbf{y}_{1:k-1}$:n suhteen, saadaan priori-jakauma:

$$p(\mathbf{x}_k | \mathbf{y}_{1:k-1}) = \int p(\mathbf{x}_k | \mathbf{x}_{k-1}) p(\mathbf{x}_{k-1} | \mathbf{y}_{1:k-1}) d\mathbf{x}_{k-1} \quad (3.4)$$

Kun ehdollistetaan tila \mathbf{x}_k mittausten $\mathbf{y}_{1:k}$:n suhteen, saadaan Bayesin kaavaa käyttäen posteriori-jakauma:

$$p(\mathbf{x}_k | \mathbf{y}_{1:k}) = \frac{p(\mathbf{y}_k | \mathbf{x}_k) p(\mathbf{x}_k | \mathbf{y}_{1:k-1})}{p(\mathbf{y}_k | \mathbf{y}_{1:k-1})} \quad (3.5)$$

3.2 Kalmanin suodatin

Kalmanin suodatin (KF, engl.: Kalman filter) perustuu bayesiläiseen suodatukseen. Kalmanin suodattimen tilamalli ja mittausmalli ovat lineaarisia.

$$\text{Alkutila} \quad \mathbf{x}_0 \quad (3.6)$$

$$\text{Tilamalli} \quad \mathbf{x}_k = \mathbf{F}_k \mathbf{x}_{k-1} + \mathbf{w}_k \quad (3.7)$$

$$\text{Mittausmalli} \quad \mathbf{y}_k = \mathbf{H}_k \mathbf{x}_k + \mathbf{v}_k \quad (3.8)$$

Kalmanin suodattimen tilaa merkitään \mathbf{x}_k . Matriisia \mathbf{F}_k kutsutaan tilansiirtomatriisiksi, ja matriisia \mathbf{H}_k mittausmatriisiksi. Alkutila noudattaa normaalijakaumaa $\mathbf{x}_0 \sim \mathcal{N}(\mathbf{x}_0, \mathbf{P}_0)$. Alkutilan odotusarvolla ja tilalla on sama merkintä, mutta ne tarkoittavat eri asioita. Virheet \mathbf{w}_k ja \mathbf{v}_k mallinnetaan valkoisella kohinalla, eli ne noudattavat gaussista jakaumaa, $\mathbf{w}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_k)$, $\mathbf{v}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_k)$, ja ovat riippumattomia. Kalmanin suodattimelle voidaan johtaa algoritmi 1. Algoritmissa \mathbf{x}_k on tilan estimaatti.

Algoritmi 1 Kalmanin suodatin [33]

Priori-estimaatit

$$\mathbf{x}_k^- = \mathbf{F}_k \mathbf{x}_{k-1}$$

$$\mathbf{P}_k^- = \mathbf{F}_k \mathbf{P}_{k-1} \mathbf{F}_k^T + \mathbf{Q}_{k-1}$$

Kalmanin vahvistus

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k)^{-1}$$

Posteriori-estimaatit

$$\mathbf{x}_k = \mathbf{x}_k^- + \mathbf{K}_k (\mathbf{y}_k - \mathbf{H}_k \mathbf{x}_k^-)$$

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_k^-$$

Kalmanin suodattimella saadut estimaatit ovat optimaalisia, sillä ne minimoivat estimaatin ja todellisen tilan keskineliövirheen.

3.3 Partikkelisuodatin

Partikkelisuodatin [14] (PF, engl.: particle filter) on bayesiläinen suodatusmenetelmä, jonka ymmärrys vaatii pienen katsauksen Monte Carlo -menetelmiin ja tärkeysotantaan.

Monte Carlo -integroinnissa pyritään ratkaisemaan funktion $g(\mathbf{x})$ integraalia.

$$I = \int g(\mathbf{x}) \, d\mathbf{x} \quad (3.9)$$

Monte Carlo -menetelmissä funktio esitetään kahden tekijän tulona $g(\mathbf{x}) = f(\mathbf{x})\pi(\mathbf{x})$, missä $\pi(\mathbf{x})$ voidaan tulkita tiheysfunktioiksi, jolloin $\pi(\mathbf{x}) \geq 0$ ja $\int \pi(\mathbf{x}) \, d\mathbf{x} = 1$. Nyt integraalin arvoa voidaan arvioida näytteistämällä jakaumasta $\pi(\cdot)$ näytteitä ja laske-
malla otoskeskiarvo kaavalla

$$I_N = \frac{1}{N} \sum_{i=1}^N f(\mathbf{x}^i) \quad (3.10)$$

Suurten lukujen lain nojalla I_N suppenee arvoon I melkein varmasti.

Tärkeysotannassa otetaan käyttöön toinen käsite, ehdokasjakauma $q(\mathbf{x})$, jolle pätee $\pi(\mathbf{x}) > 0 \Rightarrow q(\mathbf{x}) > 0$. Tämän ominaisuuden avulla yhtälö (3.9) voidaan kirjoittaa

$$I = \int f(\mathbf{x}) \frac{\pi(\mathbf{x})}{q(\mathbf{x})} q(\mathbf{x}) \, d\mathbf{x}. \quad (3.11)$$

Nyt Monte Carlon estimaatti saadaan käyttämällä jakaumasta $q(\mathbf{x})$ näytteistettyjä arvoja $\{\mathbf{x}^i\}_{i=1}^N$

$$I_N = \frac{1}{N} \sum_{i=1}^N f(\mathbf{x}^i) \tilde{w}(\mathbf{x}^i), \quad (3.12)$$

missä

$$\tilde{w}(\mathbf{x}^i) = \frac{\pi(\mathbf{x}^i)}{q(\mathbf{x}^i)}. \quad (3.13)$$

Jos tiheysfunktion $\pi(\mathbf{x})$ normalisointi ei ole tunnettu, saadaan I_N normalisoimalla painot

$$I_N = \frac{\frac{1}{N} \sum_{i=1}^N f(\mathbf{x}^i) \tilde{w}(\mathbf{x}^i)}{\frac{1}{N} \sum_{i=1}^N \tilde{w}(\mathbf{x}^i)} = \sum_{i=1}^N f(\mathbf{x}^i) w(\mathbf{x}^i), \quad (3.14)$$

missä

$$w(\mathbf{x}^i) = \frac{\tilde{w}(\mathbf{x}^i)}{\sum_{i=1}^N \tilde{w}(\mathbf{x}^i)} \quad (3.15)$$

on normalisoitu paino. Painojen normalisointia tarvitaan esimerkiksi kun $\pi(\mathbf{x})$ on posteriori-tiheys (3.5).

Kun yhdistetään kappaleen 3.1 mallit ja tärkeysotanta, saadaan perusta partikkelisuodattimelle. Yhdistetty posteriori-jakauma voidaan kirjoittaa käyttäen näytteitä

$$p(\mathbf{x}_{1:k}|\mathbf{y}_{1:k}) \approx \sum_{i=1}^N w_k^i \delta(\mathbf{x}_{1:k} - \mathbf{x}_{1:k}^i), \quad (3.16)$$

missä reaalisaatiot \mathbf{x}_k^i on näytteistetty ehdokasjakaumasta $q(\mathbf{x}_k|\mathbf{y}_k)$. Paino w_k^i heijastaa koko historiaa ja voidaan kirjoittaa

$$w_k^i \propto \frac{p(\mathbf{x}_{1:k}^i|\mathbf{y}_{1:k})}{q(\mathbf{x}_{1:k}^i|\mathbf{y}_{1:k})}. \quad (3.17)$$

Tärkeysotannan ehdokasjakauma valitaan siten, että

$$q(\mathbf{x}_{1:k}|\mathbf{y}_{1:k}) = q(\mathbf{x}_k|\mathbf{x}_{1:k-1}, \mathbf{y}_{1:k})q(\mathbf{x}_{1:k-1}|\mathbf{y}_{1:k-1}). \quad (3.18)$$

Nyt yhdistetty todennäköisyys $p(\mathbf{x}_{1:k}|\mathbf{y}_{1:k})$ esitetään sopivassa tekijämuodossa käyttäen hyväksi oletuksia (3.1) ja (3.2), sekä Bayesin lakia (2.7)

$$\begin{aligned} p(\mathbf{x}_{1:k}|\mathbf{y}_{1:k}) &= \frac{p(\mathbf{y}_k|\mathbf{x}_{1:k}, \mathbf{y}_{1:k-1})p(\mathbf{x}_{1:k}|\mathbf{y}_{1:k-1})}{p(\mathbf{y}_k|\mathbf{y}_{1:k-1})} \\ &= \frac{p(\mathbf{y}_k|\mathbf{x}_{1:k}, \mathbf{y}_{1:k-1})p(\mathbf{x}_k|\mathbf{x}_{1:k-1}, \mathbf{y}_{1:k-1})p(\mathbf{x}_{1:k-1}|\mathbf{y}_{1:k-1})}{p(\mathbf{y}_k|\mathbf{y}_{1:k-1})} \\ &= \frac{p(\mathbf{y}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{x}_{k-1})}{p(\mathbf{y}_k|\mathbf{y}_{1:k-1})}p(\mathbf{x}_{1:k-1}|\mathbf{y}_{1:k-1}) \\ &\propto p(\mathbf{y}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{x}_{k-1})p(\mathbf{x}_{1:k-1}|\mathbf{y}_{1:k-1}). \end{aligned} \quad (3.19)$$

$$\propto p(\mathbf{y}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{x}_{k-1})p(\mathbf{x}_{1:k-1}|\mathbf{y}_{1:k-1}). \quad (3.20)$$

Yhdistämällä yhtälöt (3.18) ja (3.20) yhtälöön (3.17) saadaan painoille päivitysyhtälö

$$w_k^i \propto w_{k-1}^i \frac{p(\mathbf{y}_k|\mathbf{x}_k^i)p(\mathbf{x}_k^i|\mathbf{x}_{k-1}^i)}{q(\mathbf{x}_k^i|\mathbf{x}_{k-1}^i, \mathbf{y}_k)}. \quad (3.21)$$

Posteriori-jakauma voidaan nyt siis kirjoittaa

$$p(\mathbf{x}_k|\mathbf{y}_{1:k}) \approx \sum_{i=1}^N w_k^i \delta(\mathbf{x}_k - \mathbf{x}_k^i) \quad (3.22)$$

Laskemalla päivitys (3.21) useasti peräkkäin puhutaan sekventiaalisesta painotuspoiminnasta (engl.: sequential importance sampling), joka muodostaa perustan partikkelisuodattimen algoritmille. [31]

Partikkelisuodattimessa näytteitä \mathbf{x}_k^i kutsutaan partikkeleiksi. Menetelmän avulla voidaan tarkastella epälineaarista mallia helposti, kun mallin tutkiminen suoritetaan partikkelikohtaisella simuloinnilla. Esitetyt yhtälöt olettavat vain, että todennäköisyydet

$p(\mathbf{y}_k|\mathbf{x}_k^i)$ ovat laskettavissa, tilamalli $p(\mathbf{x}_k|\mathbf{x}_{k-1})$ ja mittausmalli $p(\mathbf{y}_k|\mathbf{x}_k)$ tunnetaan, sekä uusia näytteitä voidaan näytteistää jakaumasta q .

Uudelleennäytteistys on tärkeä osa partikkelisuodattimen algoritmia. Uudelleennäytteistyksen tarkoituksena on valita partikkelit yhtä suurilla painoilla edustamaan jakaumaa, joka approksimoi edellisten partikkeleiden jakaumaa. Tämän se saavuttaa poistamalla pienipainoiset partikkelit ja tuoda ne parempaan käyttöön kopioimalla hyväpainoisten partikkelien tila usealle eri partikkelille. Ilman uudelleennäytteistystä joidenkin partikkelien paino painuu nolaksi ja partikkelit eivät enää kykene kuvaamaan estimoitavaa jakaumaa tarpeeksi hyvin. Lopulta vain yhdellä partikkelilla on merkittävästi painoa [31]. Alkuperäisessä bootstrap-suodattimessa [14] uudelleennäytteistys suoritettiin joka päivityksen jälkeen, mutta menetelmän on huomattu vähentävän partikkelien esittämän jakauman uskottavuutta.

Partikkelisuodattimia on olemassa monia variaatioita, mutta algoritmi näyttää peruseriaatteiltaan samalta. Partikkelisuodattimen alkutila saadaan, kun priorijakaumasta $p_0(\cdot)$ näytteistetään N partikkelia. Esiteltävänä algoritmina 2 on rekursiivinen päivitys partikkelisuodattimella. Ensimmäiseksi algoritmissa suoritetaan partikkelien päivitys ehdokasjakaumasta q . Usein ehdokasjakaumaksi valitaan tilamallin priorijakauma $p(\mathbf{x}_k|\mathbf{x}_{k-1})$. Tämä valinta on kuitenkin epäoptimaalinen, mutta sitä käytetään kun optimaalisen ehdokasjakauma on usein hankala tai mahdoton laskea [31]. Partikkelien painot päivitetään mittauksen uskottavuuden, tilamallin uskottavuuden ja ehdokasjakauman uskottavuuden mukaan, kuten yhtälössä 3.21. Uskottavuuden skaalauksella ei ole väliä, sillä kaikkien partikkelien paino normalisoidaan yhtälön 3.15 mukaan. Algoritmissa on mukana myös uudelleennäytteistysaskel, joka suoritetaan jos efektiivinen näytteitten määrä (ESS, engl.: effective sample size) $\hat{N}_{eff} = \frac{1}{\sum_{i=1}^N (w_k^i)^2}$ alittaa tietyn asetetun rajan.

Eräs uudelleennäytteistysalgoritmi on systemaattinen uudelleennäytteistys (algoritmi 3). Siinä painoista tehdään ikään kuin kertymä, josta otetaan tasaisin N^{-1} -välimatkoin painokertymästä nykyisen painon indeksi. Kuvassa 3.1 on havainnollistettu systemaattisen uudelleennäytteistyksen tilojen kopioimista. Tilojen kopioimisen jälkeen kaikki painot asetetaan yhtä suuriksi.

Algoritmi 2 Partikkelisuuodatin

Parametrit: Partikkelit painoineen $\{\mathbf{x}_{k-1}^i, w_{k-1}^i\}_{i=1}^N$ ja mittaus \mathbf{y}_k **Paluuarvot:** Päivitettyt partikkelit ja painot $\{\mathbf{x}_k^i, w_k^i\}_{i=1}^N$ **for** $i = 1$ **to** N **do**

$$\mathbf{x}_k^i \sim q(\mathbf{x}_k | \mathbf{x}_{k-1}^i, \mathbf{y}_k)$$

$$\tilde{w}_k^i = w_{k-1}^i \frac{p(\mathbf{y}_k | \mathbf{x}_k^i) p(\mathbf{x}_k^i | \mathbf{x}_{k-1}^i)}{q(\mathbf{x}_k^i | \mathbf{x}_{k-1}^i, \mathbf{y}_k)}$$

end for

$$t = \sum_{i=1}^N \tilde{w}_k^i$$

for $i = 1$ **to** N **do**

$$w_k^i = t^{-1} \tilde{w}_k^i$$

end for

$$\hat{N}_{eff} = \frac{1}{\sum_{i=1}^N (w_k^i)^2}$$

if $\hat{N}_{eff} < N_{raja}$ **then**

$$[\{\mathbf{x}_k^i, w_k^i\}_{i=1}^N] = \text{Systemaattinen uudelleennäytteistys}[\{\mathbf{x}_k^i, w_k^i\}_{i=1}^N]$$

end if

Algoritmi 3 Systemaattinen uudelleennäytteistys

Parametrit: Partikkelit painoineen $\{\mathbf{x}_k^i, w_k^i\}_{i=1}^N$ **Paluuarvot:** Uudelleennäytteistetyt partikkelit ja painot $\{\hat{\mathbf{x}}_k^j, w_k^j\}_{j=1}^N$

$$c_1 = w_k^1$$

for $i = 1$ **to** N **do**

$$c_i = c_{i-1} + w_k^i$$

end for

$$i = 1$$

$$u_1 \sim U(0, N^{-1})$$

for $j = 1$ **to** N **do**

$$u_j = u_1 + N^{-1}(j - 1)$$

while $u_j > c_i$ **do**

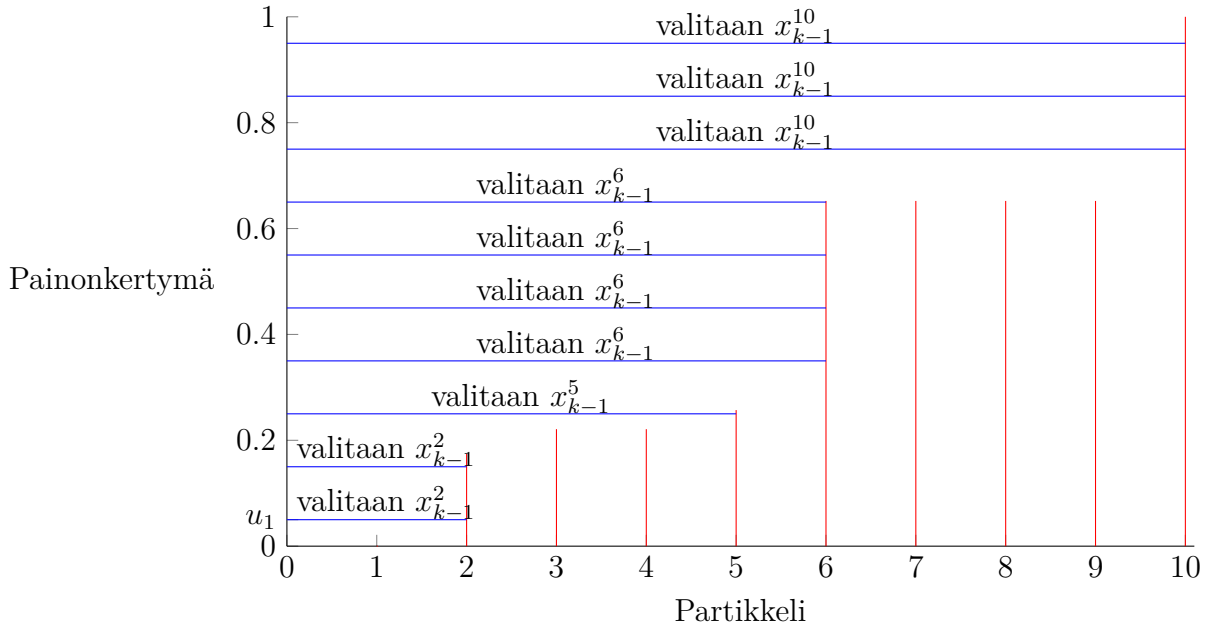
$$i = i + 1$$

end while

$$\hat{\mathbf{x}}_k^j = \mathbf{x}_k^i$$

$$w_k^j = N^{-1}$$

end for



Kuva 3.1: Systemaattisessa uudelleennäytteistyksessä partikkelien valinta. Painot ovat $\{w_k^j\}_{j=1}^{10} = (0,00; 0,17; 0,05; 0,00; 0,04; 0,40; 0,00; 0,00; 0,00; 0,35)$. Uudelleennäytteistyksen jälkeen partikkelit on kopioitu niin, että $\{\hat{\mathbf{x}}_k^j\}_{j=1}^{10} = \{\mathbf{x}_k^{a_j}\}_{j=1}^{10}$, $a = (2, 2, 5, 6, 6, 6, 6, 10, 10, 10)$.

Partikkelisuodattimen algoritmin kanssa voidaan käyttää karttatietoa. Partikkeli voidaan painottaa sen mukaan, miten ne liikkuvat kartalla. Sisätilakarttojen kanssa voidaan esimerkiksi estää yksittäisen partikkelin liikkuminen seinän läpi. [9; 35]

3.4 Työssä käytettävät mallit

Tässä työssä käytetään seuravaksi esiteltäviä malleja. Paikkaa ajanhetkellä t_k estimoidaan aikasarjassa askel-, kulma- ja WLAN-mittauksista. Tila on vektori, joka koostuu viidestä komponentista

$$\mathbf{x}_k = \begin{bmatrix} x_k \\ y_k \\ z_k \\ \alpha_k \\ l_k \end{bmatrix}, \quad (3.23)$$

missä x_k ja y_k ovat tason koordinaatteja, z_k kertoo diskreetin kerrosnumeron α_k kulman x-y -tasossa ja l_k on askeleen pituus. Nyt siis $n_x = 5$. Tason koordinaatistona käytetään rakennuksen ympärille luotoa paikallista ENU-koordinaatistoa. Koordinaatistossa yksikkö vastaa yhtä metriä, ja helpottaa mallien käyttämistä verrattuna LLH-koordinaatistoon, jossa leveys- ja pituuspiirin yksiköt ovat eri mittaisia. Tilamalli ja liikemalli ovat diskreettiaikaisia ja epälineaarisia. Tilamalli on muotoa

$$\mathbf{x}_k = f_k(\mathbf{x}_{k-1}) + \mathbf{w}_k, \quad (3.24)$$

missä $\mathbf{x}_k \in \mathbb{R}^{n_x}$ sekä $\mathbf{x}_{k-1} \in \mathbb{R}^{n_x}$ ovat vektoreita, $f_k : \mathbb{R}^{n_x} \rightarrow \mathbb{R}^{n_x}$ on tilan funktio ja \mathbf{w}_k on tilan epävarmuutta kuvaava satunnaismuuttuja. Virhevektorin komponentit voidaan olettaa likimain riippumattomiksi. Tilamalli päivitetään ajanhetkellä t_k kun k :s askel mitataan. Tilamallista saadaan ehdollinen todennäköisyys

$$p(\mathbf{x}_k | \mathbf{x}_{k-1}) = p_{\mathbf{w}_k}(\mathbf{x}_k - f_k(\mathbf{x}_{k-1})). \quad (3.25)$$

Kun tilansiirtomalli ja tilansiirtofunktio f_k kirjoitetaan auki, saadaan

$$\mathbf{x}_k = \mathbf{x}_{k-1} + \begin{bmatrix} l_{k-1} \cdot \sin(\alpha_{k-1}) \\ l_{k-1} \cdot \cos(\alpha_{k-1}) \\ 0 \\ \delta\alpha \\ 0 \end{bmatrix} + \mathbf{w}_k, \quad (3.26)$$

missä l_k on askeleen pituus, ja kulman muutos askeleelle tulee yhtälöstä $\alpha_k = \alpha_{k-1} + \delta\alpha$. Tässä $\delta\alpha$ on gyroskoopin avulla laskettu arvo kääntymiselle askeleen aikana. Kuvassa 3.2 esitetään mallin kulmat tasossa. Kuvan mukaisesti kulmaa ei ole rajoitettu tietyille välille, vaan kulma voi saada mitä vain reaaliarvoja.

Mittausmalli puolestaan on

$$\mathbf{y}_k = h_k(\mathbf{x}_k) + \mathbf{v}_k, \quad (3.27)$$

missä $\mathbf{y}_k \in \mathbb{R}^{n_y}$, $\mathbf{x}_k \in \mathbb{R}^{n_x}$ ovat vektoreita, \mathbf{v}_{k-1} on mittausmallin virhe, $h_k : \mathbb{R}^{n_x} \rightarrow \mathbb{R}^{n_y}$ mittausfunktio.

Esimerkkimittausfunktio on paikan antava mittausfunktio, joka voidaan laskea myös matriisikertolaskuna

$$\mathbf{y}_k = \mathbf{H}_k \mathbf{x}_k + \mathbf{v}_k \quad (3.28)$$

missä

$$\mathbf{H}_k = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} \quad (3.29)$$

on mittausmatriisi. Tässä työssä keskitytään paikkamittauksiin mittausmallissa. Paikan jakauma lasketaan WLAN-mittausten avulla. Tällöin virhetermi tulee myös mitauksesta, mutta se noudattaa käytetyissä menetelmissä aina jotain origokeskeistä normaalijakaumaa $\mathbf{v}_{k-1} \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_k)$

Partikkelisuodattimella voidaan käyttää myös ehdollista jakaumaa $p(\mathbf{y}_k | \mathbf{x}_k)$ suoraan. Todennäköisyys voitaisiin laskea esimerkiksi kuuluvuusaluemallin tai signaalin vaimenemismallin mukaan.

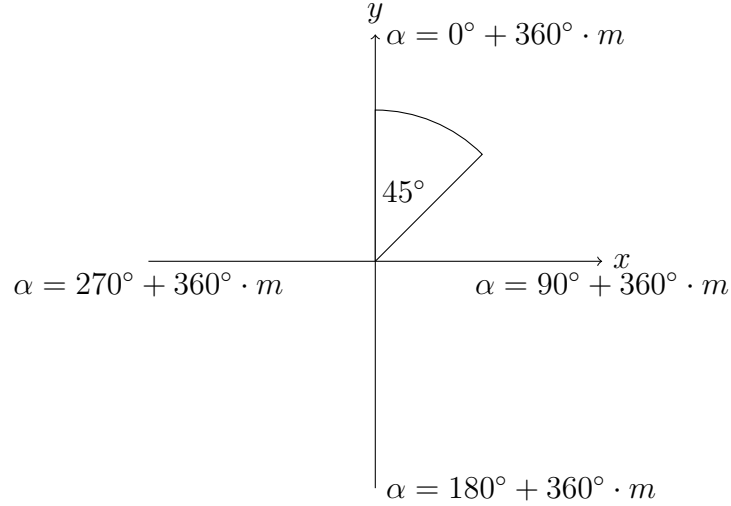
Tässäkin työssä käytetty uskottavuusfunktio on askelten ja seinien leikkausten tarkastelua tutkiva funktio:

$$\mathbf{w}_t^i \propto p_k(\mathbf{x}_t^i, \mathbf{x}_{t-1}^i) = \begin{cases} 1 & \text{kun jana } \overrightarrow{\mathbf{x}_t^i \mathbf{x}_{t-1}^i} \text{ ei leikkaa seiniä} \\ \varepsilon & \text{muuten} \end{cases} \quad (3.30)$$

Seinien läpi kävelevien partikkelien painoa siis pudotetaan. Jos $\varepsilon = 0$, partikkelit eivät vaikuta estimointiin ennen seuraavaa uudelleennäytteistystä. Myös muita uskottavuusfunktioita esitellään seuraavassa luvussa.

Jos askelmittaukset eivät ole saatavilla, voidaan 2D-liikettä mallintaa satunnaiskävelynä. Tällöin estimaatin tila sisältää vain koordinaatit.

$$\mathbf{x}_k = \begin{bmatrix} x_k \\ y_k \\ z_k \end{bmatrix} \quad (3.31)$$



Kuva 3.2: Mallin x- ja y-suunta, sekä kulma tasossa, $m \in \mathbb{Z}$

Mallin nimenkin mukaan liikkuminen on tällöin satunnaista kohinaa

$$\mathbf{x}_k = \mathbf{x}_{k-1} + \mathbf{w}_k, \quad (3.32)$$

missä \mathbf{w}_k ennustaa satunnaista liikkumista. Satunnaiskävelyn suurin ongelma on, että se ei yritä estimoida liikettä millään tavalla. Tällöin paikkaestimaattia ei voida päivittää ilman paikkamittausta.

Myös vakionopeusmalliksi kutsuttavaa liikemallia voidaan käyttää ilman askelmittauksia. Tilassa on mukana nopeus tasossa.

$$\mathbf{x}_k = \begin{bmatrix} x_k \\ y_k \\ v_{x,k} \\ v_{y,k} \end{bmatrix} \quad (3.33)$$

Paikan muutos perustuu estimoituun nopeuteen ja nopeuden muutos perustuu kohinaan.

$$\mathbf{x}_k = \mathbf{x}_{k-1} + \begin{bmatrix} v_{x,k} \\ v_{y,k} \\ 0 \\ 0 \end{bmatrix} + \mathbf{w}_{k-1} \quad (3.34)$$

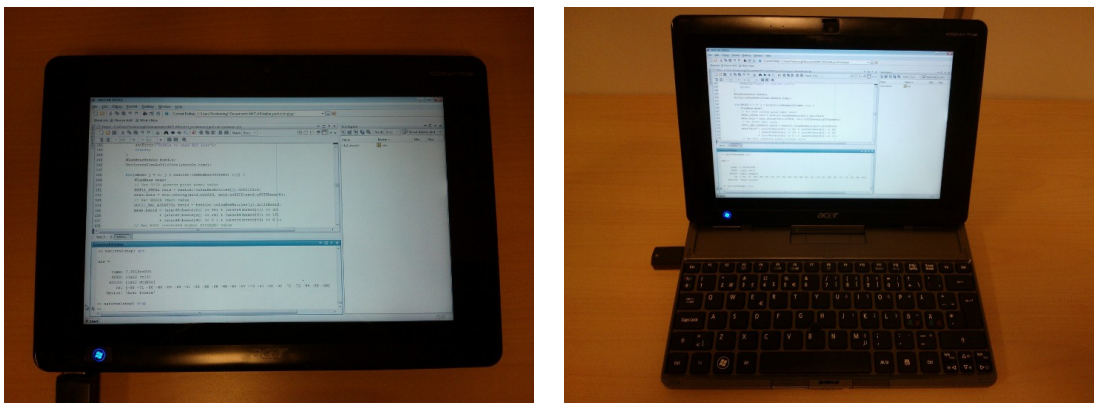
Vakionopeusmallissa voidaan virheen jakaumaa säätämällä saada aikaan nopeasti tai hitaasti muuttuvat nopeus. Vakionopeusmalli muistuttaa hyvin paljon aiemmin esiteltyä mallia askelpituudelle ja kulmalle, ja nämä mallit voidaan jopa samaistaa sopivilla muunnoksilla tilansiirtofunktioon. Kulmaa estimoiva algoritmi sopii kuitenkin paremmin, kun saadaan kulman muutos mittauksena.

4. REAALIAIKAJÄRJESTELMÄN TOTEUTTAMINEN

Eräs tämän työn tärkeistä osuuksista on reaaliaikaisen sisätilapaikannusjärjestelmän toteuttaminen. Tässä luvussa esitellään tarkemmin toteutusta. Lisäksi joitain ratkaisuja testataan ensin simulaattorien avulla.

4.1 Laitteisto

Paikannusjärjestelmä on toteutettu siten, että se toimii reaaliaikaisesti Acer Iconia Tab W500 -tietokoneella. Laite on sormitietokone, jossa on 1 gigahertsin AMD:n kaksiydinsuoritin ja 2 gigatavua RAM-muistia. Laitteen käyttöjärjestelmä on Microsoft Windows 7. Tarvittavat ohjelmistot ovat MATLAB®, ja XSens-laitteen ajurit.



Kuva 4.1: Acer W500 sormitietokone

XSens MTw on XSens-yhtiön valmistama langaton IMU-sensori. Siinä on kiihtyvyysmittari, gyroskooppi, magnetometri ja barometri. MTw kytketään tietokoneeseen USB-paikkaan kiinnitettävällä Awinda-tukiasemalla. Awinda-tukiasema yhdistetään MTw-sensoriin langattomalla yhteydellä.

Tämän työn testeissä XSens MTw oli kiinnitetty sormitietokoneen taakse. Sensorin vakiosijainti kulkemissuuntaan helpottaa askeltunnistusalgoritmin toimintaa, ja parantaa kääntymisen tarkkuutta.



Kuva 4.2: XSens MTw -sensori ja tietokoneeseen liitettävä Awinda-tukiasema

4.2 Karttainformaatio

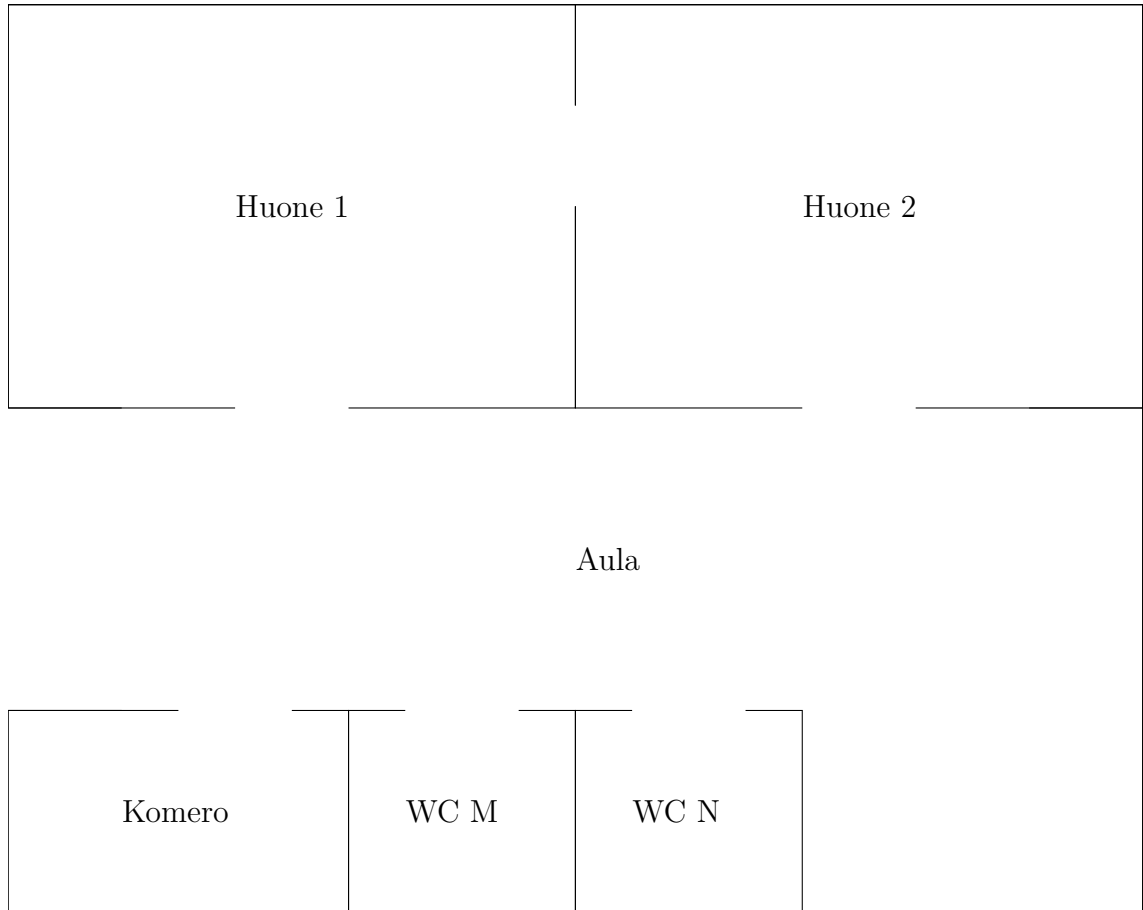
Tässä työssä käytetään Tampereen teknillisen yliopiston Sähkötalosta ja Tietotalosta muodostettuja sisätilakarttoja. Kartoissa on kuvattuna jokainen yliopistolla oleva huone tai tila erikseen huoneen nurkkien koordinaateilla. Lisäksi huoneissa voi olla ovia toisiin huoneisiin. Ovet ovat seinän päälle merkittyjä janoja. Lisäksi karttaan on merkitty, mitkä huoneet sisältävät portaikon tai hissien.

Karttatiedon kerääminen ja muodostaminen ei ole osa tätä työtä. TTY:n tilapalvelu on antanut käyttöön yliopiston tilojen kartat, joista Nokia on muodostanut Nokia Destination Maps™-kartat. Destination Maps -formaattista on saatu poimittua paikannukseen tarvittavat karttatiedot, kuten seinien ja portaiden paikat.

Kuvassa 4.3 on karttamallia esittelevä esimerkkikartta. Esimerkiksi aula-tilassa on ovi muihin näkyviin huoneisiin. Komero-tilasta on ovi vain aulaan. Mallissa seinät on yksinkertaistettu viivoiksi, vaikka todellisuudessa ne ovat paksuja.

Karttatietoa voi hyödyntää partikkelien painottamisessa usealla tavalla. Esimerkiksi opinnäytetyössä [8] seinien läpi kulkeville partikkeleille annetaan nollapaino, mutta muut partikkelit painotetaan sen mukaan, kuinka lähellä ne ovat todennäköisiä kävelyreittejä.

Tässä kappaleessa on simulaation avulla tutkittu erilaisia heuristisia karttaan perustuvia painotuksia. Tutkittavana on seinien välttämiseen perustuva painotus sekä painotus, jossa partikkelit saavat enemmän painoa kun eivät ole kulkemassa seiniä päin.



Kuva 4.3: Esimerkki karttaan sisältyvästä tiedosta

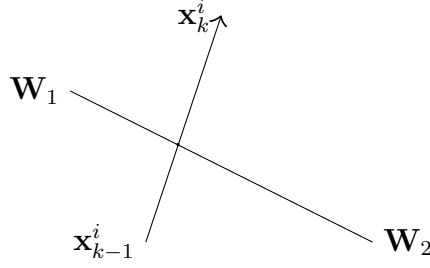
4.2.1 Partikkeliin ja seinien törmäystarkastelu

Jo aiemmin esitelty painotusmenetelmä, yhtälö (3.30), on suosittu kirjallisuudessa. Tässä työssä käytetään yleensä parametrille ε arvoa 0, mutta myös muita arvoja testataan tässä luvussa.

Seinätörmäystarkastelu onnistuu vektorilaskennalla ja janojen leikkaustestillä. Kuvasa 4.4 näytetään partikkelin tiloja \mathbf{x}_{k-1}^i ja \mathbf{x}_k^i vastaavat paikat kartalla ja niiden välille muodostettu jana sekä seinän kulmapisteet \mathbf{W}_1 ja \mathbf{W}_2 . Molemmat janat oletetaan pituudeltaan nollasta poikkeaviksi.

Ratkaisemalla yhtälö

$$\mathbf{x}_{k-1}^i + l_1(\mathbf{x}_k^i - \mathbf{x}_{k-1}^i) = \mathbf{W}_1 + l_2(\mathbf{W}_2 - \mathbf{W}_1) \quad (4.1)$$



Kuva 4.4: Törmäystarkastelun pisteet

voidaan selvittää askeleen ja seinän suuntaisten suorien leikkauspisteet. Ongelman ratkaisu voidaan muotoilla lineaarisesti yhtälöryhmäksi

$$\begin{bmatrix} \mathbf{x}_k - \mathbf{x}_{k-1} & \mathbf{W}_2 - \mathbf{W}_1 \end{bmatrix} \begin{bmatrix} l_1 \\ l_2 \end{bmatrix} = \begin{bmatrix} \mathbf{W}_1 - \mathbf{x}_{k-1} \end{bmatrix} \quad (4.2)$$

Tälle yhtälöryhmälle on olemassa yksikäsitteinen ratkaisu täsmälleen silloin, kun matriisilla $\begin{bmatrix} \mathbf{x}_k - \mathbf{x}_{k-1} & \mathbf{W}_2 - \mathbf{W}_1 \end{bmatrix}$ on olemassa käänteismatriisi. Kun matriisilla ei ole käänteismatriisia, askeleen ja seinän määrittämät suorat ovat samansuuntaiset. Jos ratkaisu on olemassa ilman käänteismatriisia, ovat suorat päällekkäiset.

Kun tiedetään 2×2 -matriisin käänteismatriisi

$$A^{-1} = \begin{bmatrix} a & b \\ c & d \end{bmatrix}^{-1} = \frac{1}{|A|} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix} = \frac{1}{ad - bc} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}, \quad (4.3)$$

voidaan muotoilla yhtälön ratkaisu

$$\begin{aligned} \begin{bmatrix} l_1 \\ l_2 \end{bmatrix} &= \frac{1}{(\mathbf{x}_{k,x} - \mathbf{x}_{k-1,x})(\mathbf{W}_{2,y} - \mathbf{W}_{1,y}) - (\mathbf{x}_{k,y} - \mathbf{x}_{k-1,y})(\mathbf{W}_{2,x} - \mathbf{W}_{1,x})} \\ &\cdot \begin{bmatrix} \mathbf{W}_{2,y} - \mathbf{W}_{1,y} & \mathbf{W}_{1,x} - \mathbf{W}_{2,x} \\ \mathbf{x}_{k-1,y} - \mathbf{x}_{k,y} & \mathbf{x}_{k,x} - \mathbf{x}_{k-1,x} \end{bmatrix} \begin{bmatrix} \mathbf{W}_{1,x} - \mathbf{x}_{k-1,x} \\ \mathbf{W}_{1,y} - \mathbf{x}_{k-1,y} \end{bmatrix} \\ &= \frac{1}{(\mathbf{x}_{k,x} - \mathbf{x}_{k-1,x})(\mathbf{W}_{2,y} - \mathbf{W}_{1,y}) - (\mathbf{x}_{k,y} - \mathbf{x}_{k-1,y})(\mathbf{W}_{2,x} - \mathbf{W}_{1,x})} \\ &\cdot \begin{bmatrix} (\mathbf{W}_{2,y} - \mathbf{W}_{1,y})(\mathbf{W}_{1,x} - \mathbf{x}_{k-1,x}) + (\mathbf{W}_{1,x} - \mathbf{W}_{2,x})(\mathbf{W}_{1,y} - \mathbf{x}_{k-1,y}) \\ (\mathbf{x}_{k-1,y} - \mathbf{x}_{k,y})(\mathbf{W}_{1,x} - \mathbf{x}_{k-1,x}) + (\mathbf{x}_{k,x} - \mathbf{x}_{k-1,x})(\mathbf{W}_{1,y} - \mathbf{x}_{k-1,y}) \end{bmatrix} \end{aligned} \quad (4.4)$$

MATLAB-kielessä ratkaisu onnistuu todella nopeasti vektoroituna laskutoimituksena. Kun arvot l_1 ja l_2 ovat väliltä $[0, 1]$, janat leikkaavat toisensa. Muussa tapauksessa leikkaus tapahtuu janan suuntaisella suoralla kuitenkin janan ulottumattomissa. Monissa ohjelmointikielissä nollalla jakaminen saattaa keskeyttää ohjelman suorituksen,

mutta MATLAB tuottaa tulokseksi liukulukustandardin *NaN*-arvon. Tämä helpottaa samansuuntaisten janojen testausta.

4.2.2 Seinien välttäminen

Eräs tutkittu painotusfunktio antaa painon partikkeleille sen mukaan kuinka lähellä ne ovat seiniä. Tutkittava yhtälö on

$$\tilde{w}_k^i \propto p(\min_j \min_{\mathbf{c} \in \overrightarrow{W_{(j)}}} d(\mathbf{x}_k^i, \mathbf{c})), \quad (4.5)$$

missä d on pisteiden välinen etäisyys, $\mathbf{c} \in \overrightarrow{W_{(j)}} = \overrightarrow{\mathbf{W}_{j_1} \mathbf{W}_{j_2}}$ jokin seinän määräämälle janalle $\overrightarrow{W_{(j)}}$ kuuluva piste, indeksi j käy läpi kaikki seinät sekä p on etäisyydestä riippuva uskottavuusfunktio.

Mallin kontekstissa partikkelia mallinnetaan pisteellä ja seinää suoralla. Pisteen etäisyys suorasta voidaan laskea helposti geometriaa käyttäen. Kuvan 4.5 avulla hahmotamme janan päätepisteet Q_1 ja Q_2 sekä pisteen P . Merkitsemme janaa $\overrightarrow{Q_1 P}$ symbolilla \mathbf{a} ja janaa $\overrightarrow{Q_1 Q_2}$ symbolilla \mathbf{b} . Projektion pituus on määritelty

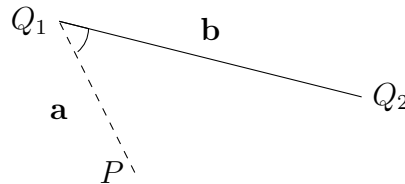
$$a_{\mathbf{b}} = \frac{\mathbf{a} \cdot \mathbf{b}}{\|\mathbf{b}\|} = \mathbf{a} \cdot \hat{\mathbf{b}} \quad (4.6)$$

Voi kuitenkin olla, että projektiovektori ei sijaitse kokonaan janalla \mathbf{b} . Näissä tapauksissa pisteen ja janan lyhin etäisyys ei ole projektio pisteessä, vaan joko janan alku tai loppupäässä. Yhteenvetona saadaan

$$\min_{\mathbf{c} \in \mathbf{b}} d(P, \mathbf{c}) = \begin{cases} \|\mathbf{a}\| & \text{jos } a_{\mathbf{b}} < 0 \\ \|\mathbf{b} - \mathbf{a}\| & \text{jos } a_{\mathbf{b}} > \|\mathbf{b}\| \\ \|a_{\mathbf{b}} \hat{\mathbf{b}} - \mathbf{a}\| & \text{muulloin} \end{cases}, \quad (4.7)$$

missä $\hat{\mathbf{b}}$ on vektorin \mathbf{b} suuntainen yksikkövektori.

Tällainen painotusjärjestelmä kaatuu helposti yksinkertaiseen heuristiikkaansa. Joka aika-askeleella laskettu painotus on riippuvainen edellisestä tilasta. Jos painotus rajoi-



Kuva 4.5: Janan ja pisteen välinen etäisyys

tetaan siten, että joka mittauksella seinän vieressä olevat partikkelit saavat vain vähän vähemmän painoa, riippuvuus muodostuu pienemmäksi ongelmaksi.

Taulukossa 4.1 on simuloidun partikkelisuodatuksen virheen tunnuslukuja muutamilla eri rajoitusarvoilla. Rajoituksella tarkoitetaan painon rajoittamista jokaista partikkelia kohden joidenkin ennalta määriteltujen arvojen väliin. Esimerkiksi rajoitus $0,4 - 1,0$ tarkoittaa, että lähempänä kuin 40 cm seinää olevat partikkelit saavat painon 0,4, ja kauempana kuin 1,0 m olevat saavat painon 1,0. Tällöin yhtälön (4.5) uskottavuusfunktio on

$$p(y) = \begin{cases} 0,4 & \text{jos } y < 0,4 \\ y & \text{jos } 0,4 \leq y \leq 1,0 \\ 1,0 & \text{jos } y > 1,0 \end{cases} \quad (4.8)$$

On huomioitavaa, että partikkelisuodatinta käytettäessä tulokset saattavat vaihdella eri testien välillä satunnaisuuden takia. Siltikin taulukon perusteella kyseinen painotusmenetelmä ei ainakaan testireitillä paranna tulosta huomattavasti, mutta sen sijaan geometrysten tarkastelujen määrä kaksinkertaistuu. Pientä parannusta estimaattiin saattaa tulla kun painoa pudotetaan vain vähän. Taulukon tulokset on laskettu 500 partikkelilla.

rajat	keskivirhe	mediaani	95%-virhe
ilman painotusta	2,63	2,38	4,49
0,0-1,0	5,94	5,95	8,91
0,4-1,0	5,96	6,05	8,85
0,7-1,0	3,23	2,92	6,04
0,9-1,0	2,31	2,23	4,19

Taulukko 4.1: Partikkelien painotus seinien etäisyyden perusteella. Virheet metreinä.

4.2.3 Seinään törmäämisen ennustaminen

Toinen painotusmenetelmä pyrkii painottamaan enemmän partikkeleita, jotka ovat liikkumassa pidempää käytävää pitkin. Tämän menetelmän mukaan partikkelit saavat painon sen mukaan, kuinka pitkä matka on edessä olevaan seinään. Kun $\mathbf{x}_k^i = [x_k^i \ y_k^i \ \alpha_k^i]^T$ on partikkelien tila, saadaan laskettua seinän W_j etäisyys soveltamalla yhtälöä (4.5) tähän tapaukseen. Sen sijaan, että otetaan partikkelien nykyinen ja seuraava sijainti, kuten yhtälössä (4.1), valitaan testattavaksi janaksi jana partikkelien nykyisestä sijainnista $\mathbf{x}' = [x_k^i \ y_k^i]^T$ ennustetun suunnan mukaiseen pisteeseen

$\mathbf{x}'' = [x_k^i \ y_k^i]^T + [\cos(\alpha_k^i) \ \sin(\alpha_k^i)]^T$. Tällöin partikkelin ja sen edessä olevan seinän etäisyydeksi määritellään

$$d_e(\mathbf{x}_k^i, W_j) = \begin{cases} l_1 & \text{jos } l_1 \geq 0 \wedge 0 \leq l_2 \leq 1 \\ \infty & \text{muulloin} \end{cases}. \quad (4.9)$$

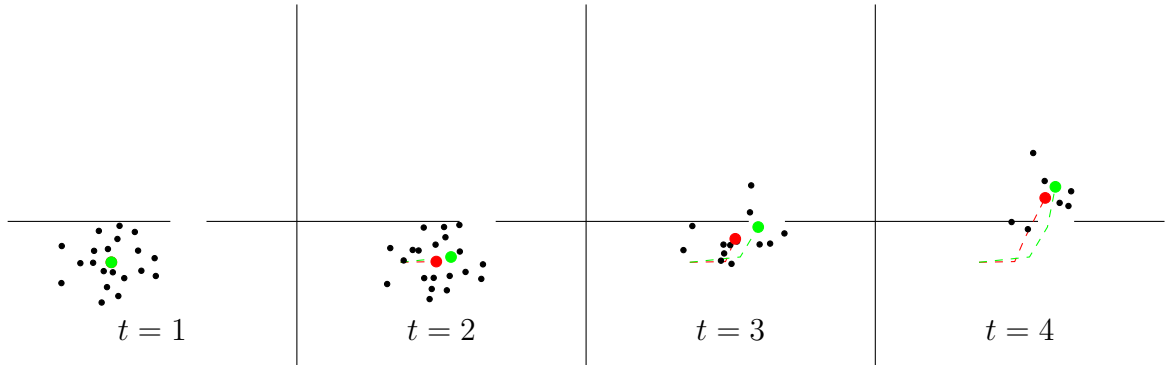
Lauseke (4.9) saa arvoksi ∞ myös silloin kun arvoja l_1 ja l_2 ei voida ratkaista singulaarisen matriisin takia.

Partikkelin paino lasketaan tällöin kaavalla

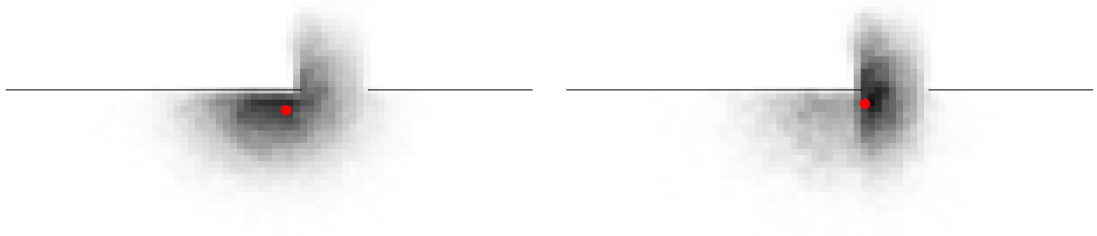
$$\tilde{w}_k^i \propto p(\min_j d_e(\mathbf{x}_k^i, W_j)), \quad (4.10)$$

missä $p(\cdot)$ on uskottavuusfunktion arvo annetusta etäisyydestä.

Menetelmä pyrkii osaltaan silottamaan tuloksia etukäteen. Kuvassa 4.6 on kuvattu tilanne, jossa partikkelien jakauma lähestyy ovea vaihtaakseen huonetta. Kun suodatin saa tiedon, että huonetta kohti on käännytty, menetelmä laskee niiden partikkelien painoa, joiden suunta osoittaa suoraan kohti seinää. Tällöin painotetun keskiarvon estimaatti siirtyy lähemmäksi oviaukkoa. Tämä vertailu on kuvassa 4.7, missä on nähtävissä partikkelimassan siirtyminen oviaukon kohdalle jo ennen seinään törmäämistä. Tuloksena suodattimen estimaatti ei ylitä seinää, vaan siirtyy siististi oviaukon kohdalta. Useimmissa testatuissa tapauksissa tällä painotuksella ei saada suurta eroa keskimääräiseen virheeseen, vaan nimenomaan estimaatin käyttäytyminen oviaukoissa paranee.



Kuva 4.6: Vihreä pallo on partikkeisuodattimen estimaatti kun painotusta (4.10) on käytetty. Punainen pallo on estimaatti kun painoon vaikuttaa vain seinään törmääminen. Simuloinnissa on mukana 20 partikkelia.



Kuva 4.7: Kuvan 4.6 kolmannen aika-askeleen partikkeleilla approksimoitu tiheysfunktio. Vasemmalla tapaus, jossa ei erityistä painoa, oikeassa kuvassa oven kohdalle siirtyneet partikkelit saavat suuremman painon. Painotettu keskiarvo on merkitty punaisella pisteellä. Estimaatin siirtyminen oven kohdalle korjaa virheellisesti seinän läpi liikkuvan estimaatin alkuperäisellä painotuksella.

4.2.4 Ihmisten käyttämien reittien suosiminen

Tutkimuksessa [8] oli toteutettu partikkelisuodatin sisätilapaikannukseen kartan avulla. Mittauksena käytettiin paitsi seinien leikkaamista, myös kartalla kulkevia todennäköisyysjanoja. Nämä janat oli sijoitettu käytävien keskellä, missä saattaa olla todennäköisintä kävellä.

Etäisyyden todennäköisyysjanoilta voi laskea soveltaen seinien välttelyyn sopivia kaavoja. Kaavaa (4.5) voidaan käyttää iteroimalla todennäköisyysjanojen yli seinien sijaan

$$\tilde{w}_k^i \propto p(\min_j \min_{\mathbf{c} \in V(j)} d(\mathbf{x}_k^i, \mathbf{c})), \quad (4.11)$$

missä V sisältää kaikki karttaan liitetyt todennäköisyysjanat. Menetelmän avulla on mahdollista rajoittaa partikkelien liikkuminen vain sallituille alueille valitsemalla yhtälön (4.11) p siten, että se pudottaa painon nopeasti kauempana sallituista alueista.

4.2.5 Esiteltyjen painotusmenetelmien vertaaminen

Taulukossa 4.2 on simulaation avulla laskettua dataa. Testissä 1000 partikkelin partikkelisuodatin päivitettiin 113 kertaa askelmittauksella ja joka viides ajanhetki WLAN-mittauksella. Molempiin mittauksiin oli lisätty virhettä ennen simulointia. Testissä alkusijainti ja suunta oletettiin tunnetuiksi. Koska partikkelisuodatin toimii joka kerta eri tavalla, testattiin jokainen menetelmä 200 kertaa.

Taulukon arvojen perusteella ei heuristisista painotusmenetelmistä ole hyötyä virheen pienentämisessä. Sitä vastoin ajoaika lähes kaksinkertaistuu ylimääräisten testien takia ja partikkelisuodatin soveltuu yhä huonommin reaaliaikaiseksi. Siksi heuristiset painotusmenetelmät eivät tule olemaan osana reaaliaikaista versiota. Referenssi-menetelmä vertaa staattisen WLAN-mittauksen virhettä todelliseen paikkaan.

menetelmä	ajokaika (s)	keskivirhe (m)	mediaanivirhe (m)	95%-kvantiilivirhe (m)
Ei mittauksia	$0,75 \pm 0,08$	$6,92 \pm 0,31$	$5,46 \pm 0,41$	$17,20 \pm 0,67$
Vain WLAN-mittaus	$0,79 \pm 0,12$	$2,71 \pm 0,10$	$2,67 \pm 0,14$	$4,44 \pm 0,15$
Seinä ja WLAN	$0,97 \pm 0,10$	$1,96 \pm 0,21$	$1,76 \pm 0,28$	$4,23 \pm 0,26$
Menetelmä 4.2.2	$1,33 \pm 0,16$	$2,07 \pm 0,31$	$1,83 \pm 0,40$	$4,58 \pm 0,32$
Menetelmä 4.2.3	$1,21 \pm 0,04$	$2,02 \pm 0,40$	$1,81 \pm 0,38$	$4,33 \pm 1,01$
Menetelmä 4.2.4	$1,52 \pm 0,02$	$1,98 \pm 0,28$	$1,78 \pm 0,36$	$4,37 \pm 0,31$
Referenssi	N/A	5,82	5,58	10,69

Taulukko 4.2: Eri mittausmenetelmien vertailua; 200 ajon keskiarvo ja otoskeskihajonta

Vaikka tässä työssä askelsuunnan virhettä käsitellään pääosin valkoisena kohinana, todellisuudessa gyroskoopin ja sitä kautta kulmamittausten aiheuttama virhe voi olla

muutakin. Eräs keino pyrkiä korjaamaan suuntaa on käyttää ihmisten suosivia reittejä priorisuuntina ja muuttaa partikkelien suuntaa näiden karttavihjeiden mukaisesti. Tässä työssä ei kuitenkaan paneuduttu liikemallin muuttamiseen, mutta menetelmästä on esimerkkinä kartalle sijoitettu graafi. [9]

4.2.6 Kartan virheet

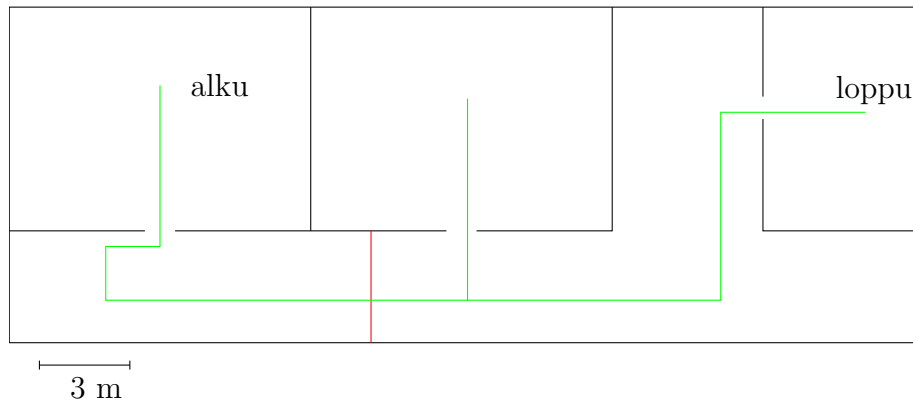
Karttatietojakaan ei voida aina olettaa täysin tarkoiksi, vaan virheitä voi esiintyä esimerkiksi puuttuvien ovien tai ylimääraisten seinien muodossa. Kartassa esiintyvät virheet saattavat johtua monesta syystä. Kartan laatija on saattanut tehdä virheen kartan koostamisessa. Pohjapiirroksen on myös voitu tehdä muutoksia kartan laatimisen jälkeen. Lisäksi kun karttaa käsitellään ohjelmalle sopivaan muotoon, siihen saattaa syntyä virheitä. Partikkelisuodattimen selviämistä kartan virheistä pyrittiin tutkimaan käyttämällä muutamaa menetelmää virheen havaitsemiseen ja korjaamiseen.

Testejä ajettiin kuvan 4.8 kartalla. Osassa testejä käytössä oli Mallin käyttäytymistä tutkittiin kaavan (3.30) uskottavuusfunktion parametrin ε eri arvoilla. Kuvassa 4.9 on simulaation avulla tutkittu hyvää arvoa parametrille ε . Parametrin edessä * tarkoittaa, että kartta ei sisällä ylimääraista punaista seinää. Kun $\varepsilon = 1$, seinät eivät käytännössä vaikuta laisinkaan suodatukseen. Menetelmä vastaa käytännössä suodattinta, jossa ei käytetä seinätörmäystestiä painotukseen. Kuvan perusteella virheet ovat samat näille menetelmille. Havaitaan, että pieni ε ei nosta virhettä paljoa. Parhaaseen tulokseen ylletään kuitenkin virheettömällä kartalla, kun $\varepsilon = 0$. Kartan virheiden takia varsinaisessa ohjelmassa valitaan nollasta poikkeava ε :n arvo.

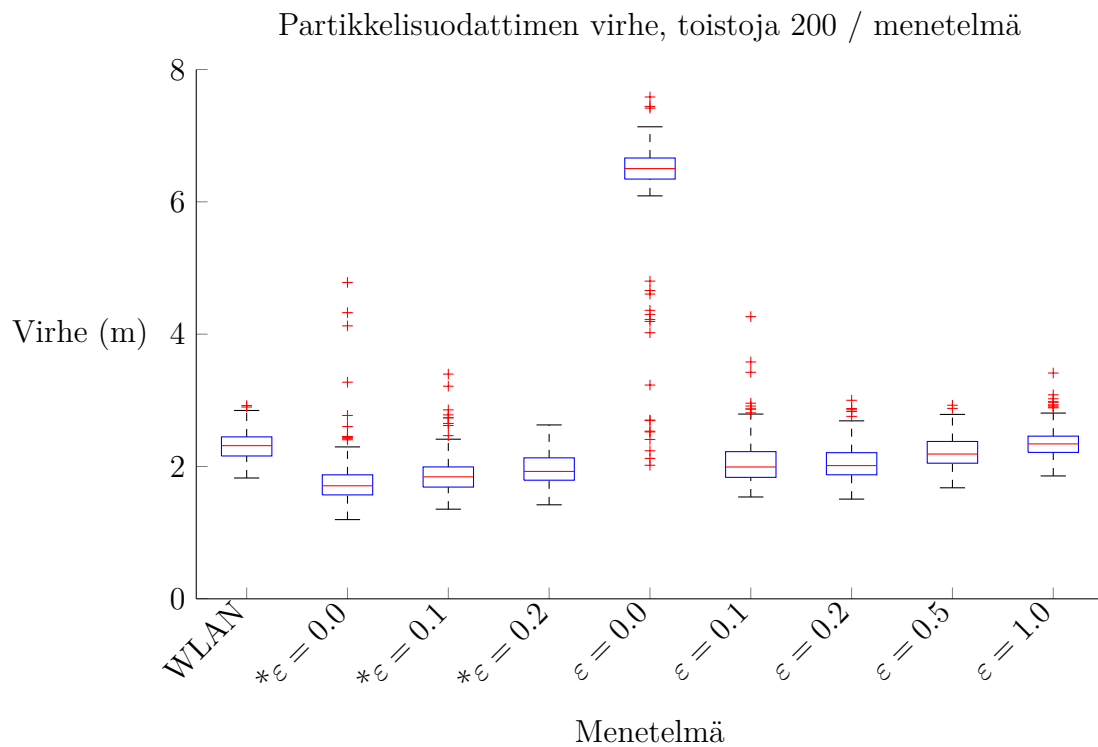
4.3 Toteutus

Järjestelmä on toteutettu pääosin MATLAB-kehitysympäristössä ja sitä käytetään sormitietokoneessa pyörivällä Microsoft Windows 7 -käyttöjärjestelmällä. Ohjelmointikielen valinta perustui pääosin MATLAB-ympäristön nopeaan ohjelmointiin ja partikkelisuodattimen helppoon toteuttamiseen tieteelliseen laskentaan tarkoitettulla ohjelmointikielellä.

Eräs ongelma toteutuksessa on ohjelman monisäikeistys. Järjestelmän on tehtävä sensori- ja WLAN-mittauksia myös silloin kun tilaestimaattia päivitetään suodattimella. MATLAB tarjoaa pääosin yksisäikeisen tulkin ohjelmia varten. Monisäikeistys on mahdollista, mutta hyvin rajoitetusti. Ohjelmoija voi pääasiassa rinnakkaistaa toistorakenteita tai funktiokutsuja. Graafista käyttöliittymää varten on myös oma säikeensä sen sulavamman toiminnan takaamiseksi, mutta tämä ei ole ohjelmoijan hallittavissa.



Kuva 4.8: Kartta, jossa testattiin karttavirheen vaikutusta. Punainen seinä oli ns. karttavirhe, sillä todellinen reitti kulki sen läpi.



Kuva 4.9: Suodattimen paikannustarkkuutta testattiin virheellisellä kartalla. Parametrin ε arvoa vaihdettiin. Tähdellä merkityissä testeissä kartassa ei ollut virheitä. Joustamaton malli, missä $\varepsilon = 0$, ei toimi virheellisellä kartalla.

MATLAB voi olla hyvin hidas paikoittain. Erityisesti for-lauseissa tapahtuva laskenta kannattaa yrittää kirjoittaa MATLABin vektoroiduilla laskutoimituksilla. Vekto-

roidussa toistossa matriisien tai vektoreiden alkioittain tapahtuvat laskuoperaatiot voidaan kirjoittaa yhdelle riville. Yksinkertaisen lausekkeen laskeminen usealle arvolla toistorakenteen avulla ohjelmakoodilistauksessa (4.1) ja vertailukohtana sama laskutoimitus vektoroituna listauksessa (4.2). Hitaus on tosin vähentynyt uusimpien MATLAB-versioiden myötä JIT-kääntämisen ansiosta.

Ohjelmakoodi 4.1: Toistorakenne

```
tulos = zeros(1000,1);
for x = 1:1000
    tulos(x) = exp(-x) * x^2;
end
```

Ohjelmakoodi 4.2: Vektorointi

```
x = 1:1000;
tulos = exp(-x) .* x.^2;
```

4.3.1 Reaaliaikaiset mittaukset

MATLAB-ympäristö ei tarjoa mahdollisuutta tehdä WLAN-mittauksia suoraan .m-funktioita käyttäen. WLAN-mittaukset on siis tehtävä joko ulkoisella ohjelmalla, tai MATLABin tukemalla natiiviksi koodiksi käännettävää C/C++-koodia tukevalla mex-rajapinnalla. Lisäksi on huomioitava, että WLAN-mittaukseen kuluu usein aikaa useampi sekunti, jolloin mittausten tekeminen täytyy tehdä samaan aikaan muun laskennan ohessa reaaliaikaisuuteen pyrittäessä.

Ensimmäinen ratkaisu oli käynnistää MATLAB-ympäristön ulkopuolelle prosessi, joka kirjoittaa tiedostoon kuultujen tukiasemien BSSID-tunnisteet ja RSS-arvot. Ratkaisu oli hidas tiedoston käsittelyn takia, ja tiedoston lukeminen ja kirjoittaminen kahdesta prosessista aiheutti välillä ongelmia. Toteutin Windows 7 -käyttöjärjestelmän wlanapi-kirjaston kanssa toimivan C++/Mex-ohjelman, joka mittaa WLAN-tukiasemien signaalien voimakkuutta. WLAN-mittaukset tapahtuvat täydellisen taka-alalla, ja MATLABin suoritussäie voi jatkaa toimintaansa häiriöttä.

WLAN-mittausten hyödyntämiseksi on ensin kerättävä tietoa radioympäristöstä käyttäjän paikannusta varten. Tässä työssä sekä kalibrointi että paikannus suoritetaan samalla laitteella. Eri WLAN-laitteet voivat kuitenkin raportoida erisuuruisia RSS-arvoja [19], mutta tämä on jätetty tässä työssä huomioimatta.

WLAN-radiokartat kerättiin Acer-taulutietokoneelle tehdyllä ohjelmalla, jossa sormitietokoneen näytöllä olevaa karttaa koskettamalla valittiin sijainti, ja sen jälkeen suoritettiin WLAN-mittaus napin painalluksella.

Tukiasemia mitattiin sekä Tampereen teknillisen yliopiston Tietotalossa että Sähkötalossa. Tietotalossa jokaisessa mittauspisteessä suoritettiin kolme peräkkäistä WLAN-mittausta ja sähkötalossa yksi. Tietokantaan tallennetaan mittauksessa kuultujen tu-

kiasemien BSSID/MAC-tunnus, signaalin voimakkuus (RSSI), sekä mittauksen kellonaika ja käsin valittu sijainti.

Kerätylle WLAN-radiokartalle on estimoitu tukiasemien parametrit käyttäen julkaisussa [27] esiteltyjä menetelmiä. Tässä kuvattu keräys suoritettiin siis offline-tilassa, jolloin suodattimella on jo tiedossaan WLAN-radioverkolle ominaiset parametrit käyttäjän paikannusvaiheessa.

Paikannusvaiheessa lasketaan saaduista mittauksista posteriori-estimaattia paikan jakaumalle kaavasta (2.16). Laskenta suoritettiin hilamenetelmällä. Prioriksi oletettiin rakennuksen kokoinen tasajakautunut hila. Käytetty rakennus oletettiin tunnetuksi. Tällöin posterioriestimaatti voi muodostua lähes minne tahansa rakennuksen alueelle.

Sensorimittaukset kerätään XSens MTw -mittauslaitteen avulla. Laite on hollantilaisen XSens Technologies -yhtiön kehittämä pieni sensoriteknologiaa sisältävä laite, joka yhdistetään tietokoneeseen langattomasti XSens-yhtiön Awinda-radioprotokollalla. Mittauslaitteessa on kiihtyvyysanturi, gyroskooppi, magnetometri ja barometri ja se kykenee keräämään dataa jopa 120 hertsin taajuudella.

Laitteen mukana toimitetaan ohjelmistokehitykseen vaadittavat komponentit ja joi-tain esimerkkikoodeja. Mukana on myös rajapinta MATLABin kautta käytettäväksi. Rajapinta toteuttaa 512-mittauksen puskurin automaattisesti. Se tarkoittaa, että 100 hertsin mittauksilla riittää lukea puskuri noin viiden sekunnin välein.

Järjestelmää varten toteutettiin yksinkertainen käyttöliittymä sensoreihin. Mittaus-ten tekeminen onnistuu suodatuskoodista *start*, *get* ja *stop* -aliohjelmien avulla. Nämä aliohjelmat käyttävät XSens-laitteiden natiivia rajapintaa Windowsin COM-rajapinnan avulla suoraan MATLAB-koodista.

Kun ohjelma kutsuu *start*-aliohjelmaa, alkaa yhteyden muodostaminen ja tietokoneeseen kiinnitettyssä Awinda-tukiasemassa alkaa vilkkua valo. Tällöin XSens MTw -mittauslaitetta on liikutettava, jotta se menee päälle ja alkaa etsiä tukiasemaa. Myös sen valo alkaa vilkkua. Kun XSens MTw -laitteen ja Awinda-tukiaseman valot vilkkuvat samaan aikaan, on yhteys muodostettu.

Mittaukset kertyvät automaattisesti mittauslaitteen puskuriin. Mittausten lukeminen *get*-aliohjelmaa kutsumalla tyhjentää puskurin. Laite lisää mittauksiin ID-numerot, jolloin ohjelma voi havaita puuttuvat mittaukset. Mittauksia saattaa puuttua esimerkiksi jos käytettävällä radiotaajuudella esiintyy häiriöitä.

Mittaus lopetetaan *stop*-aliohjelmaa kutsumalla. Tällöin mittauslaite lopettaa mit-tausten lukemisen ja palaa normaalitilaan. Mittausten lopettaminen toimii hyvin jos ohjelman suorituksessa ei tapahdu virheitä. Moneen kertaan kehityksen aikana sattui

kuitenkin virheitä ohjelmassa ja järjestelmä saattoi kadottaa kahvan. Tällöin XSens-laitteet on nollattava.

Sensorimittauksista askeleet laskeva komponentti on ohjelmassa erillään partikkelisuodattimesta. Tämä sallii sen, että partikkelisuodatin saa syötteenään suoraan askeleet ja sensoridatan prosessointi jää omalle aliohjelmalleen. Tällöin myös on mahdollista ohjelmoida partikkelisuodatin toimimaan ilman sensorimittauksia käyttäen satunnaiskävelyä tai muuta vastaavaa. Askeleentunnistusalgoritmi tutkii kiihtyvyyden normin muutosta. Koska painovoima on mukana mitatussa kiihtyvyydessä, algoritmin 4 kiihtyvyyden rajoiksi valittiin $r_{ylä} = 11 \text{ m/s}^2$ ja $r_{ala} = 9,5 \text{ m/s}^2$.

Algoritmi 4 Askeleentunnistus

Parametrit: Kiihtyvyydet sisältävä matriisi A

Paluuarvot: Havaitut askeleet

```

askeleet  $\leftarrow$  [ ]
for  $i = 1$  to  $m$  do
   $a \leftarrow \|A_{(i)}\|$ 
  if  $a > r_{ylä}$  and  $nyt - \text{askel.negaika} < t_{raja}$  then
    askel.neg  $\leftarrow false$ 
    askeleet[ ]  $\leftarrow i$ 
  end if
  if  $a < r_{ala}$  then
    askel.neg  $\leftarrow true$ 
    askel.negaika  $\leftarrow nyt$ 
  end if
end for

```

Askeleiden suunnan muutos lasketaan sensorin gyro-mittauksista. Mittauslaitteen orientaatiota arvioidaan kiihtyvyyden perusteella. Pienen ajan kiihtyvyysdata keskiarvoistetaan ja lopputuloksen suuntaista vektoria käytetään mittauslaitteen alaspäin osoittavana suuntana. Kunhan mittauslaitetta ei pyöritellä, asettuu alaspäin osoittava suunta suurinpiirtein oikeaan suuntaan lyhyelläkin keskiarvoistuksella.

4.3.2 Partikkelien mallinnus

Partikkelien mallinnuksen pohjana käytetään kappaleen 3.4 malleja. Malleja on kuitenkin sovellettava hieman simulaatiota varten.

Kun partikkelisuodattimen mittauksina toimivat sensoreiden mittauksista tuotetut estimaatit askeleen käännöskulmalle $\delta\alpha_t$, voidaan partikkeleiden liikettä mallintaa seuraavalla mallilla

$$\mathbf{x}_t^i = \begin{bmatrix} x_t^i \\ y_t^i \\ z_t^i \\ \alpha_t^i \\ l_t^i \end{bmatrix} = \begin{bmatrix} x_{t-1}^i + l_t^i \sin(\alpha_t^i) \\ y_{t-1}^i + l_t^i \cos(\alpha_t^i) \\ z_{t-1}^i \\ \alpha_{t-1}^i + \delta\alpha_t^i \\ l_{t-1}^i + \delta l_t^i \end{bmatrix}, \quad (4.12)$$

missä $\delta\alpha_t^i \sim N(\delta\alpha_t, \sigma_{t,2}^2)$. Kulmanmittausvirheen keskihajonnaksi $\sigma_{t,2}$ on tämän luvun testeissä valittu $1,5^\circ$. Uusi askel näytteistetään käyttäen $\delta l_t^i \sim N(0, 1 \text{ cm}, (0,5 \text{ cm})^2)$. Nollasta poikkeava odotusarvo auttaa partikkeleita pitämään askeleen pidempänä, sillä pitempiä askeleita ottavat partikkelit törmäävät keskimäärin nopeammin seinään. Lisäksi askelpituus on rajoitettu 60 cm ja 80 cm välille.

Simuloinnin aloitus vaatii aloitustilan. Tämä aloitustila saadaan ottamalla näyteittä priori-jakaumasta.

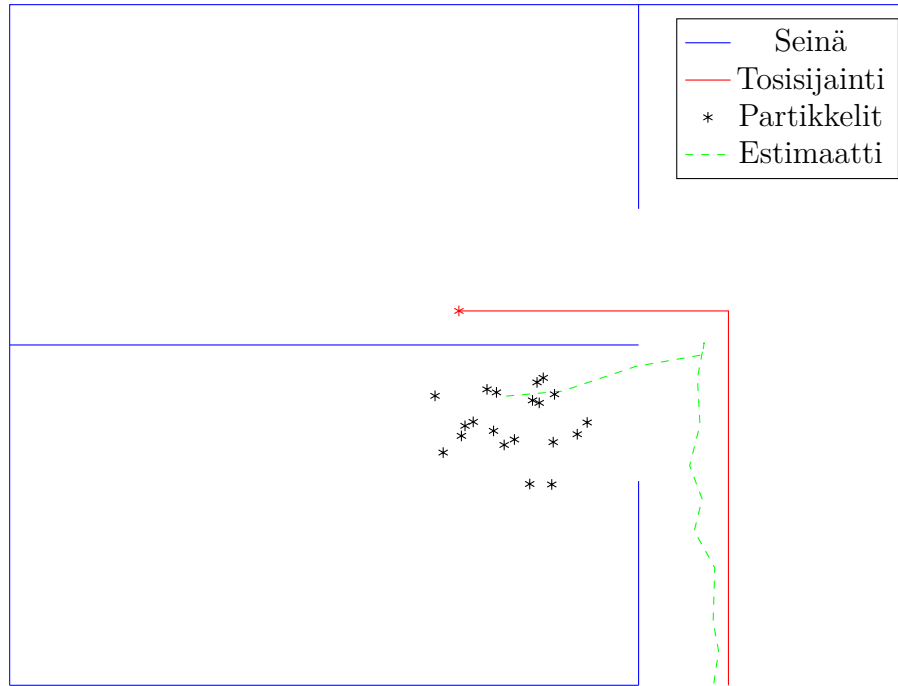
Jos tiedetään tarkka aloituspaikka, voidaan tätä käyttää priori-jakauman odotusarvona. Tällöin paikan epävarmuudeksi riittää pieni varianssi. Testeissä on käytössä tarkka arvio oikeasta aloituspaikasta pienellä varianssilla.

Tässä työssä oletetaan todella hyvän WLAN-mittauksen tai GPS-mittauksen tarkkuudella saatu aloituspaikka simulaatiolle. Tällöin käytettävien partikkelien määrä voidaan pitää suhteellisen pienenä verrattuna yhden rakennuksen kokoiseen prioriiin.

Partikkelien malli ei kata kaikkea, vaan suodatin voi myös joutua karille. Tämä tarkoittaa, että syystä tai toisesta simuloidut partikkelit eivät ennusta enää todellista posteriori-jakaumaa. Jumiutumiseen voi johtaa esimerkiksi suodattimen huono käynnistyminen. Jos tarpeeksi tarkkaa priorisijaintia ei ole saatavilla, mittauksen perusteella tehtävä alustaminen voi epäonnistua. Sisätiloissa on myös usein tapauksia, joissa partikkelijakauma saattaa päätyä väärään tilaan, kun käytävät tai huoneet ovat lähellä toisiaan. Tällöin tässä työssä esitelty suodatin saattaa valita väärän käytävän, ilman mahdollisuutta palata oikealle reitille. [32]

Tyypillinen jumiutumiseen johtava tapaus on esitelty kuvassa 4.10. Väärä tila voidaan valita esimerkiksi hieman virheellisen WLAN-mittauksen perusteella, joka painottaa väärässä huoneessa, kuitenkin vain pienen välimatkan päässä olevia partikkeleita. Myös puuttuvat tai ylimääräiset askelmittaukset aiheuttavat helposti tällaisen virheen.

Suodattimen virhetilojen havaitsemista on tutkittu huomattavan vähän muuhun tutkimukseen nähden. Jumiutumisen yhteydessä suodatin on uudelleenkäynnistettävä uudelleenalustuksen sijasta. Tällöin partikkelien tila on uudelleenalustettava joko mittauksesta tai viimeisimmästä hyvästä tilasta, jos suodattimen tilahistoriaa voidaan säi-



Kuva 4.10: Kuvassa partikkelit valitsevat väärän huoneen.

lyttää. Eräs ehdotettu kriteeri on Kullback-Leibler-divergenssin laskeminen. [32] Arvo mittaa käytännössä kahden todennäköisyysjakaman yhtenevyyttä, tässä tapauksessa todennäköisyysjakaumaksi tulkitun mittauksen ja partikkelipilvestä koostetun posteriorijakauman yhtenevyyttä. Kullback-Leibler-divergenssiä ei voida laskea, mikäli mittauksesta ei voida laskea jakaumaa, kuten suoraan signaalin vaimenemismallista laskemalla.

Suodattimen jumiutumista pyritään välttämään käyttämällä kaavan (3.30) parametrilla ε nollaa suurempaa arvoa. Tässä työssä partikkelisuodattimen uudelleenkäynnistykseen käytetään yksinkertaisesti WLAN-mittausta. Jos havaitaan, että partikkeleiden painosta alle 0,1% on WLAN-mittauksen 95%-kuuluvuusellipsin alueella, alustetaan partikkelit WLAN-mittauksen osoittamaan paikkaan. Tämä menetelmä tuo suodatukseen joustavuutta, mutta on herkkä virheellisille mittauksille.

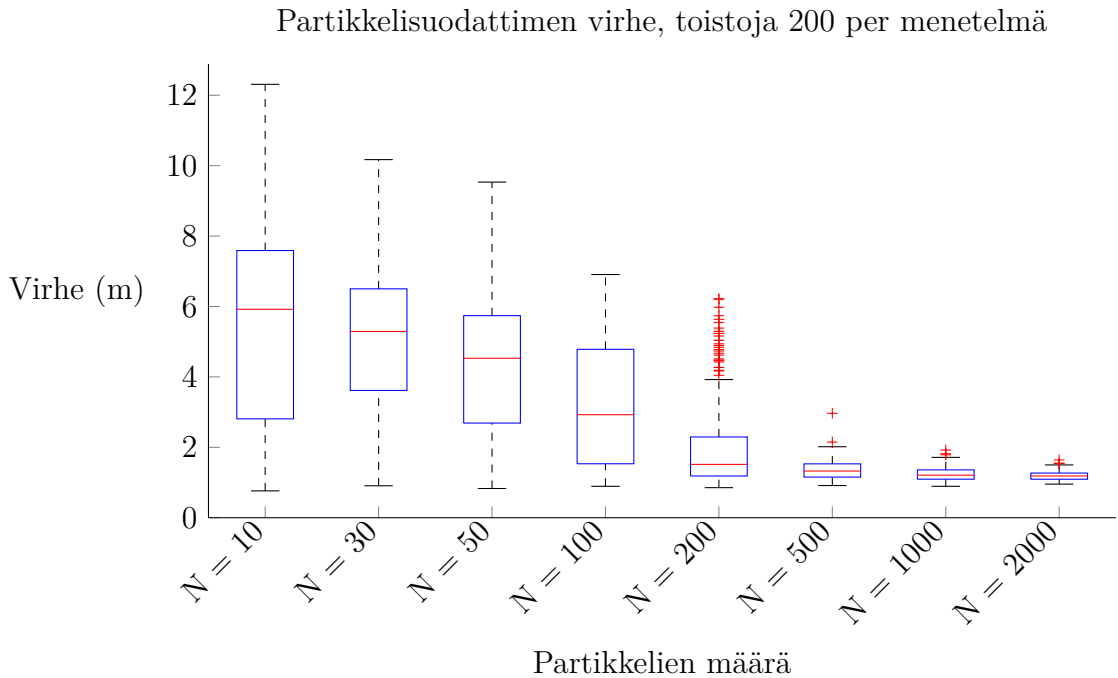
4.3.3 Tarvittava partikkelien määrä

Reaaliaikaisen partikkelisuodattimen kunnollisen toiminnan kannalta partikkelien määrä on optimoitava sopivaksi. Suodatin ei saa käyttää liikaa laskenta-aikaa yhden mittauksen prosessointiin. Toisaalta liian pieni partikkelien määrä saattaa laskea partikkelien esittämän jakauman hyvyttä. Partikkelisuodattimen teorian mukaan partikkelien määrittämä jakauma lähestyy oikeaa posteriori-jakaumaa kun partikkelien

määrä lähestyy ääretöntä [7]. Työssä [37] on testattu tasaista priori-jakaumaa suurelle toimistorakennukselle. Suppenemiseen tarvittavien partikkelien määrä on yli neljä miljoonaa. Yksinkertaisella WLAN-mittaukseen perustuvalla laajalla priorilla paikannukseen tarvittavien partikkeleiden määrä tippui noin kolmaskymmenesosaan.

Simulaation avulla tutkittiin partikkelisuodattimen toimintaa eri partikkelimäärillä. Mittauksina käytettiin WLAN-mittauksia ja seinään törmätessä partikkelin paino pudotettiin nollaan.

Kuvan 4.11 laatikko-viiksi-kaaviossa esitetään keskivirheen muutos eri partikkelimäärillä. Kaaviosta voi havaita, että virhe painuu sitä alemmas, mitä enemmän on partikkeleita. Koska kaaviot on piirretty simulaation pohjalta, todellisissa mittaukset ja kartta voivat olla epätarkempia ja partikkeleita tarvitaan eri määrä tietyn tarkkuuden saavuttamiseksi. Simuloinnin perusteella hyvillä mittauksilla parin metrin tarkkuus voitaisiin saavuttaa jo noin 500 partikkelilla.



Kuva 4.11: Suodattimen virhe partikkelien määrän funktiona. Laatikko-viiksi-kaaviossa laatikon yläraja on virheen 75%-kvantiili, alaraja 25%-kvantiili, viikset tavalinen vaihteluväli, sekä mediaanivirhe punaiselle keskellä merkittynä. Ulkolaiset, tässä tapauksessa noin 2,7 päässä keskiarvosta olevat estimaatit, piirretään erikseen.

4.3.4 Ohjelman optimointi

Reaaliaikatavoitteiden saavuttamiseksi ohjelman koodia oli optimoitava nopeaksi. MATLAB-ohjelmiston sisältämä profilointityökalu auttoi ohjelman pullonkaulojen löy-

tämisessä. Ohjelman kehityksen aikana pullonkauloja löytyi useita. Muistin varaaminen toistorakenteissa hidastaa ohjelmaa merkittävästi. Tällöin MATLAB joutuu jatkuvasti varaamaan lisää muistia. Lisäksi tietorakenteen varaama muisti ei ole yksi lohko tietokoneen muistissa, vaan sijoiteltu useampaan paikkaan. Tällöin matriisin käyttäminen hidastuu. [17] Muistinvaraus on pääosin hoidettava ennen toistorakennetta. Muistinvarauksesta esimerkki ohjelmakoodissa 4.2.

Partikkelisuodattimen hitain yksittäinen pullonkaula on aliohjelma, jossa lasketaan karttaan perustuvat mittaukset. Geometristen törmäystarkastusten määrä nousee partikkelien määrän noustessa. Jotta seinätestausten määrä voitaisiin pitää mahdollisimman pienenä, rakennettiin erityinen tietorakenne seinien säilömistä varten. Seinät tallennetaan kolmiulotteiseen solutaulukkoon. Yksi indeksi kuvaa kerrosta, ja kaksi muuta jakaa kerroksen hilaan. Hilan koko on määritelty etukäteen, ja se on sama koko alueella. Jokaisen solutaulukon soluun tallennetaan ne seinät, jotka leikkaavat solutaulukon solua vastaavaa hilan kartalla. Toteutettu seinänleikkaustestaus on esitetty algoritmossa 5. Ohjelmalistauksen rivin 3 aliohjelma toteutettiin C++/Mex-ohjelmalla. Tätä aliohjelmaa kutsutaan joka partikkelille joka päivituksen jälkeen. Tästä syystä sen on toimittava mahdollisimman nopeasti.

Algoritmi 5 Partikkelien ja seinien testaus

Parametrit: Partikkelien edellinen ja nykyinen tila $\{\mathbf{x}_{k-1}^i\}_{i=1}^N$, $\{\mathbf{x}_k^i\}_{i=1}^N$, Seinähila

Paluuarvot: Uskottavuus w

for $i = 1$ **to** N **do**

$I_z :=$ partikkelin \mathbf{x}_k^i kerros

$\{(I_{x,j}, I_{y,j})\}_{j=1}^M :=$ hilan indeksit partikkelin liikkeelle $\overrightarrow{\mathbf{x}_{k-1}^i \mathbf{x}_k^i}$

for $j = 1$ **to** M **do**

 Seinät $:=$ Seinähila $\{I_z, I_{x,j}, I_{y,j}\}$

 Laske törmäystestit käyttäen (4.5) joka seinälle

 Painota $w_i \propto \varepsilon$, jos liike leikkasi seinää

end for

end for

Piirtäminen on myös raskasta MATLABissa. Suodattimen toimintaa voi visualisoida piirtämällä partikkelipilven ja partikkelipilvestä lasketun paikkaestimaatin. Lisäksi WLAN-mittausten visualisointi helpottaa suodattimen toiminnan seuraamista. Nämä asiat on piirrettävä kartan päälle ollakseen mielekkäitä. Eri asioitten piirtämistä testattiin paljon, ja lopulta päädyttiin piirtämään seinät, partikkelipilven painotetun keskiarvon estimaatti sekä estimoitu reitti. Reaaliaikatilassa kaikkien partikkelien piirtäminen osoittautui liian raskaaksi, mutta offline-tilassa on mahdollista piirtää kaikki.

Käytännössä piirtämisestä syntyneet pullonkaulat on mahdollista poistaa käyttämällä tehokkaampaa graafista ympäristöä. Törmäystarkastelu voitaisiin sen yksinkertaisen laskennan ansiosta suorittaa myös näytönohjaimilla, jotka ovat erikoistu-

neet rinnakkaiseen laskentaan. MATLAB tukee myös CUDA-rinnakkaistusta NVIDIA-näytönohjaimella, mutta käytetyssä tietokoneessa ei ole yhteensopivaa näytönohjainta.

4.3.5 Ohjelman käyttöliittymä ja rakenne

Suodatinta varten on rakennettu graafinen käyttöliittymä. Käyttöliittymästä voidaan valita suodattimeen ja tuloksen esittämiseen liittyviä asetuksia. Lisäksi käyttöliittymästä on mahdollista syöttää suodattimelle priori-sijainnin odotusarvo ja kovarianssi. Käyttöliittymästä voidaan valita aiemmin mitatut mittaukset ja ajaa suodatin niiden pohjalta, tai valita reaaliaikaila, jossa mitataan mittauksia ja ajetaan suodatinta samaan aikaan. Ohjelman käyttöliittymä on esillä kuvassa 4.12. Käyttöliittymä on englanninkielinen.

Algoritmissa 6 kuvataan pääohjelman rakenne. Jos ohjelma pyörii offline-tilassa, sensorit toimivat virtuaalisesti. Reaaliaikaisen toteutuksen kannalta on tärkeää jatkuvasti päivittää mittauksia ja ohjelman sisäistä tilaa. Tällöin mittauksia ei pääse kertymään useita. Oletuksena ohjelma lukee mittaukset puolen sekunnin välein. Partikkelisuodattimen päivitys saattaa kestää kauemmin, mikäli edelliseen puolen sekunnin ikkunaan mahtui useita mittauksia. Keskimäärin ohjelma selviää kuitenkin normaalimäärästä askelia tarpeeksi nopeasti.

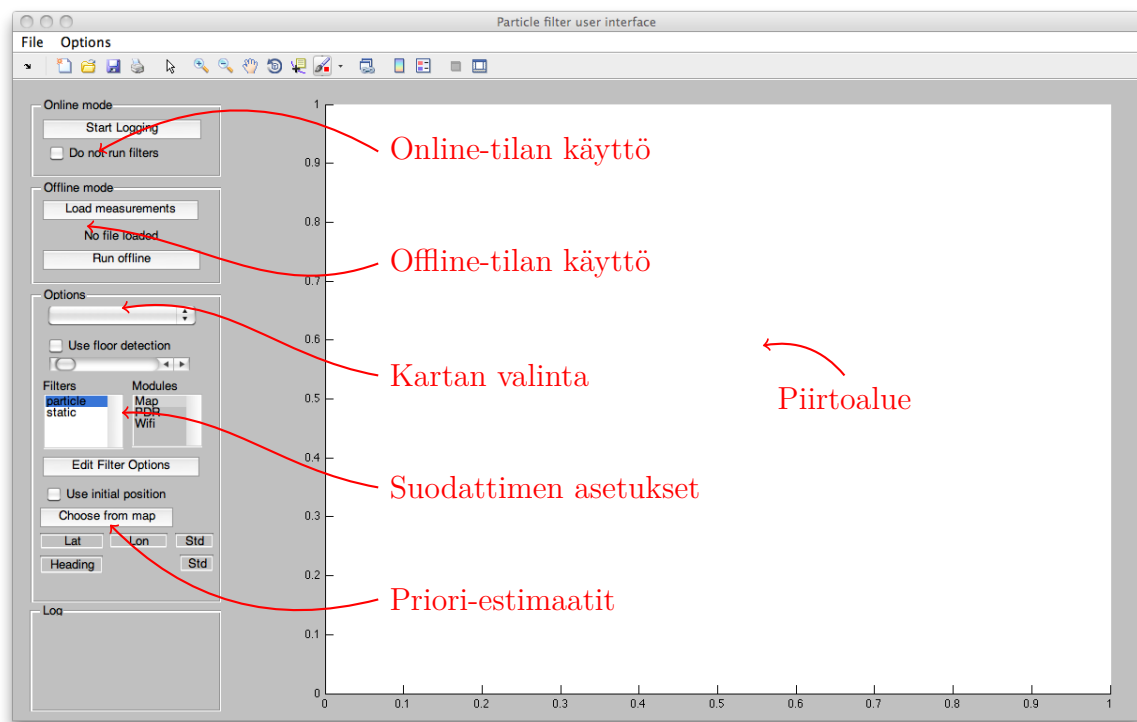
MATLABin toteutuksen takia ohjelman käyttöliittymä ei toimi sulavasti suodatusprosessin aikana. Samaa aikaan kun partikkelisuodatin laskee, käyttöliittymästä ei voida vaihtaa piirtoasetuksia. Partikkelisuodatinta päivitetään kaksi kertaa sekunnissa uusilla mittauksilla. Jos uutta WLAN-mittausta tai askelmittausta ei ole saatu, suodatin ei tee mitään. Suodattimen päivityksen jälkeen ohjelma prosessoi mahdolliset tapahtumat käyttöliittymässä.

4.4 Testaustulokset

Reaaliaikajärjestelmää testattiin todellisella mittauksilla. Testit suoritettiin Tampereen teknillisessä yliopistossa. Ennen reaaliaikatestausta testiympäristöstä kerättiin sormenjäljet WLAN-ympäristön parametrien määrittämiseen.

4.4.1 Testidatan mittaaminen

Testidata kerätään paikannusohjelmassa reaaliaikailassa. Tällöin laite kerää mittauksia ja mahdollisesti suorittaa suodatusalgoritmeja. Näyttöä painamalla käyttäjä voi valita sen hetkisen sijaintinsa kartalta. Tällöin reitille voidaan laskea virhe. Virheenlaskentaa varten askeleet interpoloidaan käyttäjän valitsemien sijaintien välille. Mittaukset ovat käytettävissä heti reaaliaikaisesti suodattimelle, mutta ne tallennetaan



Kuva 4.12: Ohjelman graafinen käyttöliittymä

Algoritmi 6 Pääohjelma

```

PF.tila := alusta.partikkelisuodatin()
PDR.tila := alusta.PDR()
aloita.wlan()
aloita.sensori()
while mittaukset jatkuvat do
   $y_{WLAN}$  := lue.wlan()
   $y_{sensori}$  := lue.sensori()
  (PDR.tila,  $y_{askel}$ ) := päivitä.PDR(PDR.tila,  $y_{sensori}$ )
  PF.tila := päivitä.partikkelisuodatin(PF.tila,  $y_{WLAN}$ ,  $y_{askel}$ )
  tallenna.mittaukset( $y_{WLAN}$ ,  $y_{sensori}$ )
  piirrä.estimaatit(PF.tila)
  prosessoi.tapahtumat()
end while
lopetta.wlan()
lopetta.sensori()
  
```

myös mittaustietokantaan offline-paikannusta varten. Ohjelma liittää yhteen samalla ajanhetkellä saadut sensori- ja WLAN-mittaukset. Tällöin reitti on mahdollista suorittaa samoja mittauksia käyttäen uudelleen. Testattavia reittejä tallennettiin useita tulosten tarkempaa tarkastelua varten. Tapa, jolla tosipaikka valitaan kartalta, saattaa aiheuttaa muutamien metrien virheen. Laitteen 10"näytöllä sormenpään alle mahtuu noin 10 metriä leveä alue yliopiston kartalla. Selvät virheet korjattiin ennen testien ajamista.

Ohjelman käyttöliittymästä voidaan myös valita, ajetaanko samalla partikkelisuodattinta, ja mitä piirretään näytölle. Piirrettäväksi voidaan valita estimoitu paikka, paikan kovarianssi, estimoitu reitti, WLAN-mittauksen kovarianssi ja partikkelien sijainti. MATLAB-ympäristössä piirtäminen on hyvin hidasta ja parhaan reaaliaikaestimaatin saavuttamiseksi vain estimoitua sijaintia päivitettiin reaaliaikaisesti. Offline-tilassa testatessa partikkelipilvi voidaan piirtää ongelmitta, kun reaaliaikavaatimusta ei ole enää.

4.4.2 Tulokset

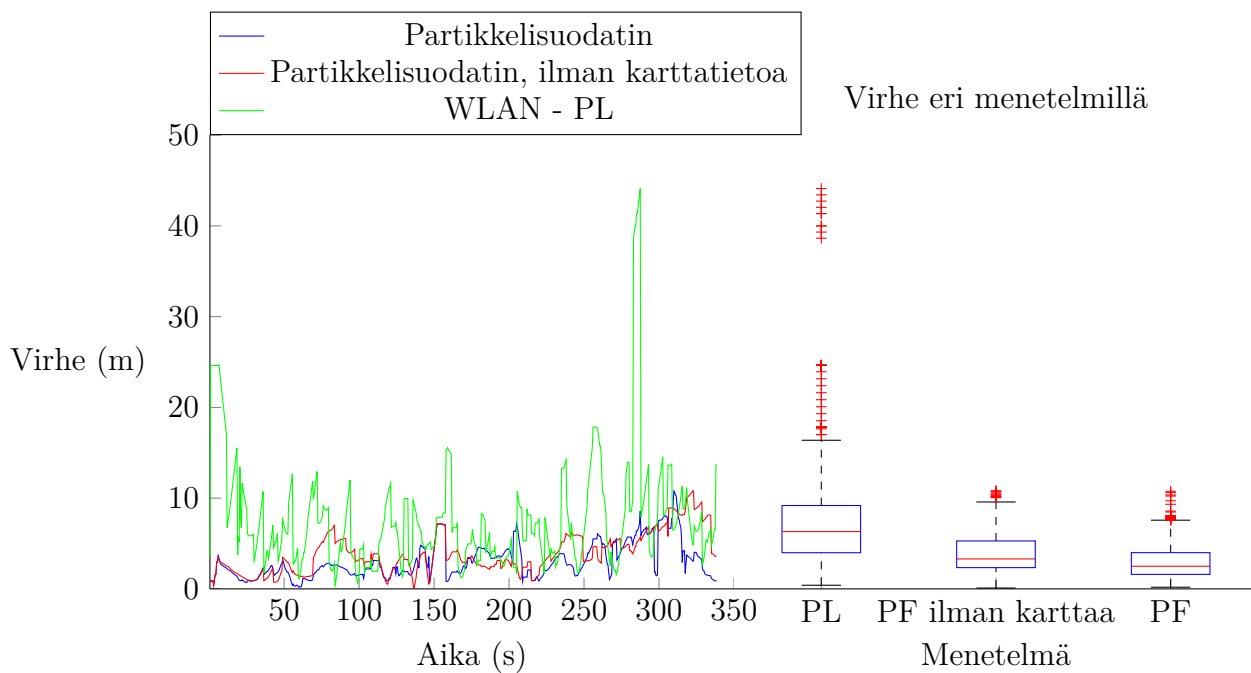
Ensimmäinen testireitti sijoittuu Tampereen teknillisen yliopiston Tietotalon pääkerrokseen. Reitti kulkee aluksi rakennuksen pääkäytäviä pitkin, mutta poikkeaa myös luentosaliin ja toimistokäytävään. Partikkelisuodatin alustetaan priori-jakaumalla, jonka odotusarvo on asetettu oikeaan paikkaan ja kovarianssi on 2 m^2 . Myös oikea aloitussuunta on annettu 10 asteen keskihajonnan priorina. Vertailun vuoksi samalle testidatalle ajetaan partikkelisuodatin sekä karttatiedon kanssa että ilman karttatietoa. Karttatiedosta on huomioitu seiniin törmääminen, mikä pudottaa painon kymmenesosaan. Testireitti sekä suodattimien ja staattisen WLAN-ratkaisijan antamat estimaatit ovat esillä kuvassa 4.13. Kyseisen testireitin virhe reitin eri paikannuksen vaiheissa on esitelty kuvassa 4.14. Taulukosta 4.3 voidaan nähdä, että partikkelisuodatin saa staattista WLAN-paikannusta paremman estimaatin lähes koko reitin ajan. Lisäksi karttatiedollinen partikkelisuodatin voittaa karttatiedottoman lähes koko reitin ajan. WLAN-paikannusta testattiin sekä kuuluvuusaluemallilla (CA), että signaalin vaimenemismallilla (PL). Testeissä havaittiin vaimenemismalli hieman tarkemmaksi, ja sitä käytettiin partikkelisuodattimen mittauksiin. Kuuluvuusaluemallin estimaattia ei piirretty kuviin. Myös Kalmanin suodattimella testattiin paikannusta käyttäen satunnaiskävelyn mallia, mutta tulos pysyi lähes samana kuin staattisessa WLAN-paikannuksessa.

	menetelmä	keskivirhe	medianivirhe	90%-virhe	95%-virhe
	Partikkelisuodatin	3,05	2,49	6,21	7,14
	Partikkelisuodatin, ei karttatietoa	3,88	3,30	7,16	8,57
	Kalmanin suodatin	7,62	6,40	13,56	16,56
	WLAN - PL	7,68	6,31	13,72	17,82
	WLAN - CA	11,93	10,59	20,18	23,72

Taulukko 4.3: Tietotalon testireitin virheen tunnuslukuja (m).



Kuva 4.13: Tietotalon testireitti. Suurempi kuva liitteessä A. kuvassa A.2

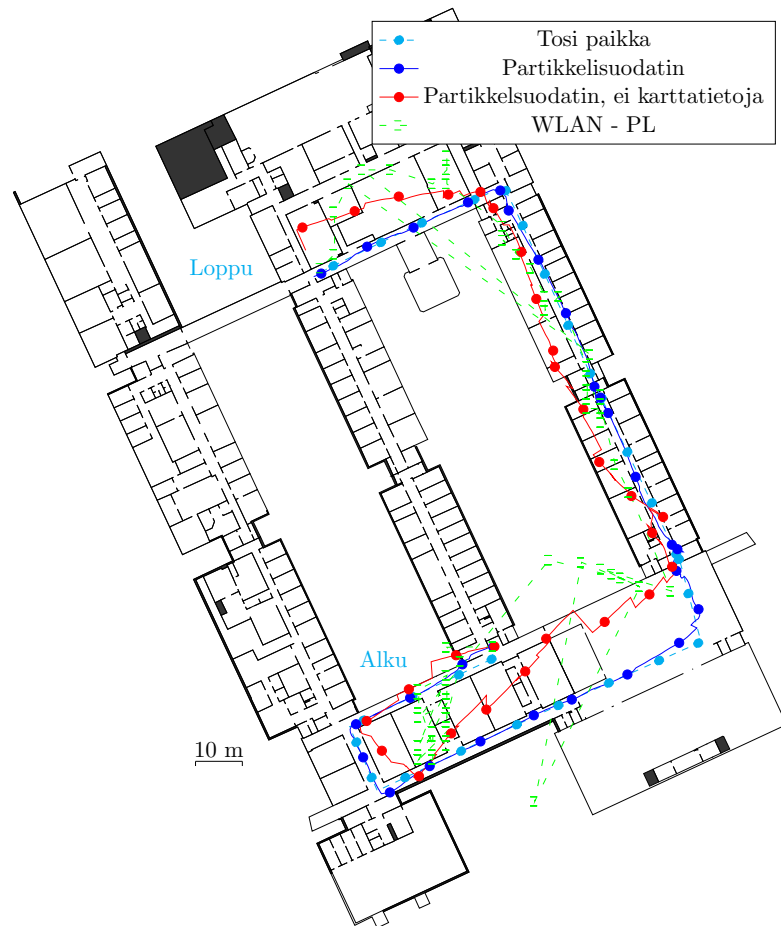


Kuva 4.14: Tietotalon testireitin virhe ajan mukaan ja virheen laatikko-viiksi-kaavio. Partikkelisuodatin, jossa karttaa ei käytetä, estimoii lähinnä saatuja askelia.

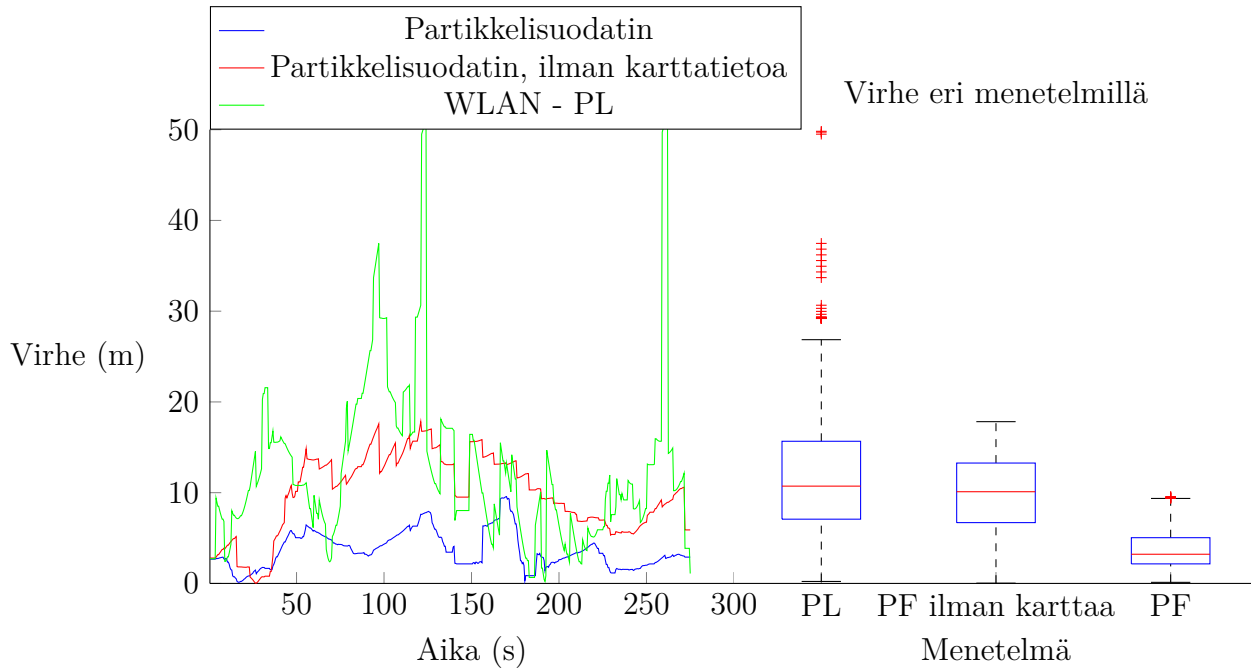
Toinen testireitti sijaitsee TTY:n Sähkötalon toisessa kerroksessa. Testireitti ja eri menetelmien estimaatit on piirretty kuvaan 4.15. Tietotaloon verrattuna reitin WLAN-mittauksista huonoimmat ovat paljon huonompia. Erityisesti kuuluvuusaluepaikannus johti huonoihin tuloksiin sähkötalossa. Tämä selittyy osittain epätasaisesti jaetulla radiokartalla, jolloin tukiasemat paikantuivat tiettyihin pisteisiin helpommin. Silti partikkelisuodattimen avulla myös Sähkötalon reitillä saavutetaan lähes sama tarkkuus kuin Tietotalon testireitillä, kuten taulukosta 4.4 voi havaita.

	menetelmä	keskivirhe	medianivirhe	90%-virhe	95%-virhe
	Partikkelisuodatin	3,64	3,22	6,35	7,59
	Partikkelisuodatin, ei karttatietoa	9,80	10,10	15,31	15,85
	Kalmanin suodatin	12,74	11,05	25,69	30,52
	WLAN - PL	12,46	10,72	21,52	29,29
	WLAN - CA	21,02	17,93	41,90	52,80

Taulukko 4.4: Sähkötalon testireitin virheen tunnuslukuja (m)



Kuva 4.15: Sähkötalon testireitti. Suurempi versio kuvasta löytyy liitteestä A. kuvasta A.1



Kuva 4.16: Sähkötalon testireitin virhe ajan mukaan ja virheen laatikko-viiksi-kaavio.

4.4.3 Tulosten analysointia

Edellä esitetyt tulokset vastaavat tarkkuudeltaan monia kirjallisuudessa esiteltyjä tuloksia. Vaikka joissain lähteissä sisätilapaikannuksessa on saatu virhe keskimäärin 1-2 metrin tarkkuudelle partikkelisuodatinta käyttäen, oletukset ovat näissä monesti paremmat. Esimerkiksi WLAN-paikannuksen tarkkuutta voidaan parantaa käyttämällä varta vasten testausta varten rakennettua verkkoa, jolloin tukiasemien tarkat sijainnit ovat tiedossa. Vielä parempi tulos PDR-reitille saadaan kiinnittämällä sensorit jalkaan, jolloin askelpituudelle voidaan saada parempi arvio. Jotkut tutkimukset käyttävät magnetometrin avulla kompassia suunnan määrittämiseen gyroskoopin sijaan. Tässä työssä ei haluttu kuitenkaan kiinnittää inertiaalisensorin asentoa karttaan nähdessä, vaan gyroskoopin tarjoamaa vapaata pyörimistä omassa koordinaatistossa.

Voidaan havaita, että todellisella datalla virheet ovat suurempia kuin simuloituissa tapauksissa. Tämä johtunee sekä pienemmistä virheistä simuloituissa tapauksissa että huonommin ennustettavasti virheestä. Todellisiin mittauksiin perustuvassa suodattimessa virheiden kovarianssien väärin arvaaminen aiheuttaa partikkelien simuloinnin väärästä jakaumasta. Joko liian iso tai liian pieni kovarianssi heikentää helposti partikkelisuodattimen estimaattia.

Sähkötalossa kuuluvuusaluemallipaikannus johti huonompaan tuloksiin kuin muut mallit. Tämä on selitettävissä osittain puutteellisen opetusdatan perusteella. Kuulu-

vuusaluemallissa oletetaan tasaisesti jakautunut opetusdata, ja sähkötalossa opetusdata ei ole jakautunut tasaisesti. Esimerkiksi jotkin käytävät on kielletty ulkopuolisilta ja niistä ei ole lainkaan mittauksia. Tällaisessa tilanteessa kuuluvuusalueet saattavat muodostua epätasaisesti.

5. MONEN KERROKSEN ONGELMA

Tässä luvussa esitellään monessa kerroksessa toimivaa partikkelisuodatinta. Edelleen partikkelisuodattimen alkutilan jakauma on tunnettu ja tarkka, mutta malliin lisätään mahdollisuus vaihtaa kerrosta.

5.1 Mallin muutokset

Karttatieto sisältää tiedon siitä, missä tiloissa on mahdollista kulkea hissin tai portaikon kautta toiseen tilaan. Tällaista huonetta voidaan kutsua yhteyshuoneeksi. Partikkelisuodattimen malliin on lisättävä kerrosten vaihtaminen. Oletetaan, että saatava mittaus $d_{z,k} \in \{-1, 0, 1\}$ kertoo kerroksen muutoksen ajanhetkellä t_k , ja z_k on ajanhetken t_k estimoitu kerros. Tällöin mallinnamme kerroksen muutosta uskottavuudella

$$p(\mathbf{x}_k | \mathbf{x}_{k-1}) \propto \begin{cases} 1 & \text{jos } z_k - z_{k-1} = d_z \wedge C(\mathbf{x}_k, d_z) \\ \varepsilon_{\text{kerrosvaihto}} & \text{muuten,} \end{cases} \quad (5.1)$$

missä $C(\mathbf{x}, d)$ on tosi, jos tilan \mathbf{x} osoittamasta rakennuksen alueesta on yhteys toiseen alueeseen, joka on d :n kerroksen päässä. Vastaava malli on artikkelissa [37], missä jalkaan kiinnitetyn kiihtyvyysanturin kanssa estimoidaan korkeussuunnassa muutosta askeleen aikana. Artikkelissa kartta on mallinnettu portaan tarkkuudella. Tällaisessa mallissa uskottavuus suppenee portaikossa nopeasti.

Partikkelisuodattimella kerrosta voidaan estimoida laskemalla kussakin kerroksessa olevien partikkelien painot yhteen yhtälöllä

$$p(z_k | \{\mathbf{x}_k^i\}_{i=1}^N) = \frac{\sum_{i \in I_{z_k}} w_k^i}{\sum_{i=1}^N w_k^i}, \quad (5.2)$$

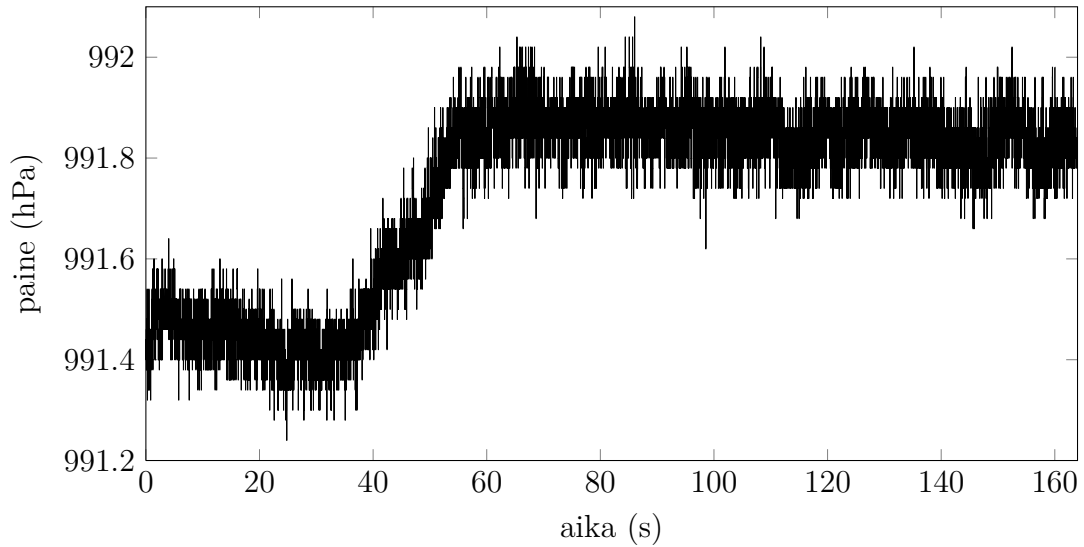
missä $I_{z_k} = \{i \mid \mathbf{x}_k^i \text{ on kerroksessa } z_k\}$.

Tässä esiteltävää lisäystä malliin voidaan soveltaa helposti portaikkoon, mutta hississä menetelmällä on vaikea tunnistaa, kuinka monta kerrosta mahdollisesti siirryttiin.

5.2 Barometrin mittauksen käyttäminen

Kerroksen vaihtumisen tunnistamiseen käytetään XSens-laitteen sisältämää barometriä. Barometrin ulostulo on ilmanpaine hehtopascalissa. TTY:n tietotalon pääkerroksessa ilmanpaine on noin 991,9hPa.

Barometrin ulostulon signaali värähtelee nopealla taajuudella. Ihmisilmällä signaalista pystyy havaitsemaan helposti jo muutaman metrin muutoksen aiheuttaman paineen muutoksen. Kuvan 5.1 oleva signaali on mitattu reitillä, jolla korkeus laski noin viisi metriä. Ilmanpaine on suoraan verrannollinen mittauslaitteen yläpuolella olevaan massaan, joten korkeuden laskiessa ilmamassan määrä ja massa kasvaa ja näin ollen ilmanpaine kasvaa.



Kuva 5.1: Ilmanpainemittauksia kahden kerroksen testireitiltä

Signaalin odotusarvon muutosta voidaan testata esimerkiksi CUSUM-menetelmällä [1]. Menetelmässä lasketaan mittausten residuaalin kumulatiivista summaa, ja verrataan tätä tehtävälle asetettuun vakioparametriin. CUSUM-algoritmissa myös rajoitetaan summa nollan jommallekummalle puolelle liian heilahtelun estämiseksi. Jotta sekä positiiviseen että negatiiviseen suuntaan voidaan tarkastella muutosta, tarvitaan kaksi summaa.

Tehtävää varten CUSUM-algoritmia on muokattu siten, että saaduista signaalimitauksista lasketaan tietyssä ikkunassa keskiarvo, jota vähennetään mittauksesta residuaalin saamiseksi. Algoritmissa 7 esitetään kokonaisuudessaan kerroksenvaihdon tunnistamiseksi algoritmi. Säästöparametri h on asetettu testissä arvoon 30, joka havaittiin Sähkötalossa suoritetuissa testeissä hyväksi.

Algoritmi 7 Adaptiivinen CUSUM(\mathbf{y})

```

 $s_1 = 0$ 
 $s_2 = 0$ 
for  $i = W$  to  $N$  do
   $\nu = \frac{1}{W} \sum_{j=i-W}^i y_j$ 
   $s_1 = \max(0, s_1 + y_i - \nu)$ 
   $s_2 = \min(0, s_2 + y_i - \nu)$ 
  if  $s_1 > h$  then
    Siirtymä alaspäin
     $s_1 = 0$ 
     $s_2 = 0$ 
  end if
  if  $s_2 < -h$  then
    Siirtymä ylöspäin
     $s_1 = 0$ 
     $s_2 = 0$ 
  end if
end for

```

5.3 Testireitti

Monen kerroksen paikannusta testataan Tietotalossa mitatulla testireitillä, joka kulkee kakkoskerroksesta portaikon kautta ykköskerroksen pääkäytävään. Testireitti on esitetty kuvassa 5.2. Tässä testissä niille partikkeleille, jotka eivät vaihda kerrosta $\varepsilon_{\text{kerrosvaihto}}$ on 0,0001.

Taulukossa 5.1 on testireitin virhe karttaa käyttävälle partikkelisuodattimelle. Päinvastoin kuin edellisessä luvussa, monen kerroksen testissä ei ole mielekästä testata

karttaa käyttämätöntä partikkelisuodatinta. Virhe on tason virhe, eli mahdollista kerrosvirhettä ei oteta siinä huomioon. Mallissa on diskreetti kerros, joten on mielekästä sanoa, että estimaatti on oikein, kunhan se vaihtuu portaissa kävelyn aikana.

Kuvasta 5.4 nähdään partikkelien ennustamat kerrokset ajon eri vaiheissa. Kuvassa nähtävä ykkös- ja kakkoskerroksen todennäköisyyksien hyppääminen vastakkaisiksi tapahtuu oikeassa paikassa. Suurin osa partikkeleista, jotka eivät vaihtaneet kerrosta, menettää painonsa pian törmättyään seiniin. Kuitenkin ennen kuin kaikki partikkelit valitsevat saman kerroksen, tapahtuu uudelleennäytteistys muutaman kerran, jolloin eniten painoa saanut kerros vaihtuu. Jos kerroksen vaihdon kriteeriksi valitaan 80% partikkelien massasta uudessa kerroksessa, on reitin estimointi oikein koko matkan jaan. Jos kriteeriksi valitaan partikkelien enimmistö, oikea kerros on valittu lähes joka ajanhetki.

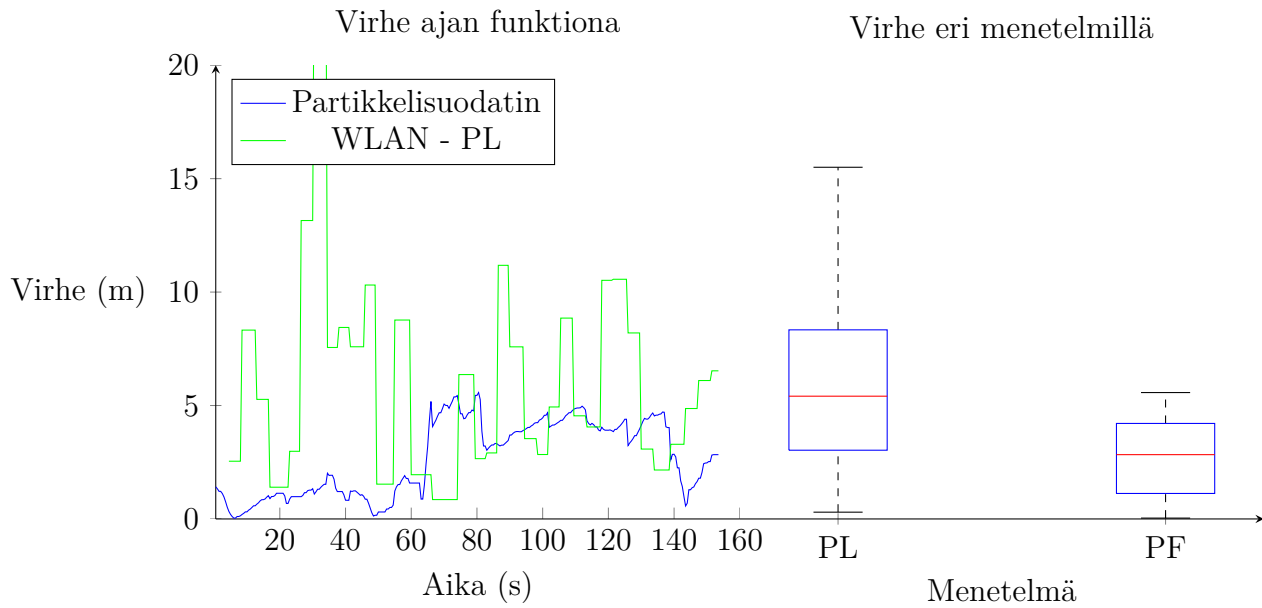
Reitille on laskettu myös konsistenttius. Paikkaestimaatin sanotaan olevan konsistentti, jos tosi paikka on reitin tietyn virhe-ellipsin sisällä. Taulukon 5.1 konsistenttiuslaskelma ei ota huomioon mahdollista virhettä kerroksessa. Mittauksen sanotaan olevan 95%-konsistentti, jos todellisen paikka mahtuu estimaatin 95%-virhe-ellipsin sisään. Taulukon sarakkeessa oleva luku tarkoittaa sitä estimaattien osuutta mittaussarjassa, joissa virhe-estimaatti oli konsistentti. Partikkelisuodattimen hieman liian pieni konsistenttius saattaa johtua väärin arvioituista jakaumista askelmittauksissa.

menetelmä	keskivirhe	mediaanivirhe	90%-virhe	95%-virhe	95%-konsistenttius
Partikkelisuodatin	2,66	2,83	4,69	4,92	81%
WLAN - PL	6,22	5,41	10,16	14,70	100%

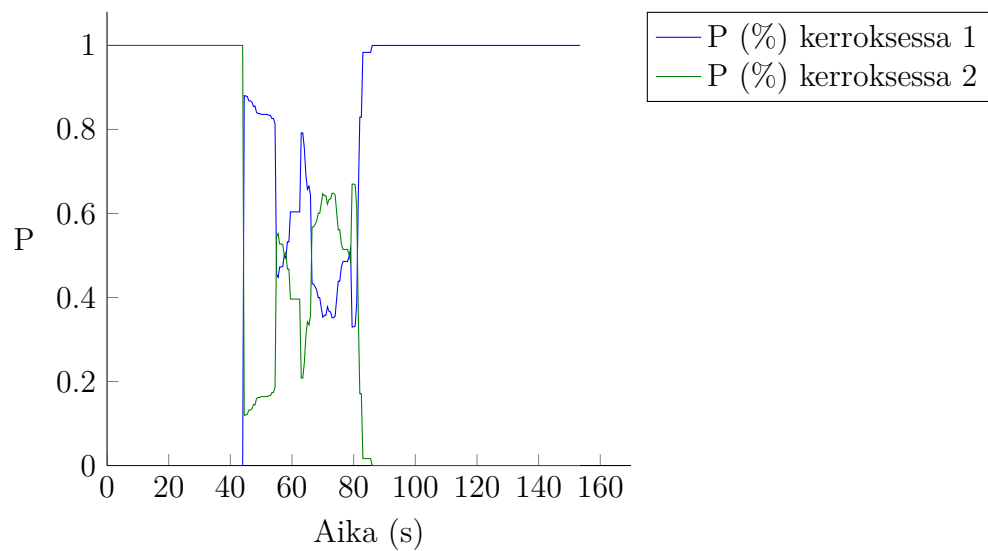
Taulukko 5.1: Kaksikerroksisen testireitin virheen tunnuslukuja (m)



Kuva 5.2: Kaksikerroksinen testireitti. Suurempi versio kuvasta löytyy liitteestä A. kuvasta A.3



Kuva 5.3: Testireitin virhe ajan mukaan ja virheen laatikko-viiksi-kaavio. Huomattavaa on, että vertailukohteena oleva WLAN-mittaus ei pysty ennustamaan kerrosta.



Kuva 5.4: Eri kerroksien todennäköisyys reitin eri vaiheissa

6. JOHTOPÄÄTÖKSET

Tässä diplomityössä on esitetty reaaliaikaisesti toimiva sisätilapaikannusjärjestelmä. Paikannus on toteutettu käyttäen inertiaalisensoreiden mittauksia ja WLAN-mittauksia yhdessä partikkelisuodattimen kanssa. Eräs tutkittu seikka on rakennusten pohjapiirrosten käyttäminen yhdessä partikkelisuodattimen kanssa. Simulaattorien avulla tutkittiin erilaisia karttaan perustuvia painotusmenetelmiä partikkelisuodattimen kanssa.

Simulaattorien tulosten perusteella panostettiin kartan käyttöön yksinkertaisesti niin, että uskottavuusfunktio antoi heikomman uskottavuuden seinien läpi kävelyille tai kartalta ulos kävelyyn. Säästynyt laskenta-aika käytettiin lisäämällä partikkelien määrää reaaliaikaisessa paikannusjärjestelmässä. Alustavissa kokeiluissa paikannusjärjestelmällä päästiin jopa noin 3 metrin mediaanivirheeseen Tampereen teknillisessä yliopistossa suoritetuissa testeissä. Tulos on samaa luokkaa muihin samankaltaisiin testeihin verrattuna [8; 35]. Partikkelisuodatinta testattiin myös kahden kerroksen testireitillä hyvin tuloksin.

Tässä työssä suodatusta pohdittiin pääosin tilanteessa, jossa alkusijainnin priorijakama oli hyvin tarkka. Alkutilan valitseminen ilman käyttäjän syötettä on paljon vaikeampi tehtävä. Eräs alkutilan valinnan pääongelmista on kerroksen tunnistaminen, jotta oikean kerroksen pohjapiirrosta voidaan käyttää estimaatin laskemiseen tai visualisointiin [36]. Kerrostunnistusta on tutkittu jo paljon WLAN-tukiasemien avulla eri menetelmillä [10]. Kun oikea kerros voidaan paikallistaa hyvällä todennäköisyydellä, reaaliaikajärjestelmän käyttömahdollisuudet monipuolistuvat. Osittain myös partikkeleita voidaan käyttää kerroksen tunnistamiseen tässä työssä esiteltyin menetelmin.

KIRJALLISUUTTA

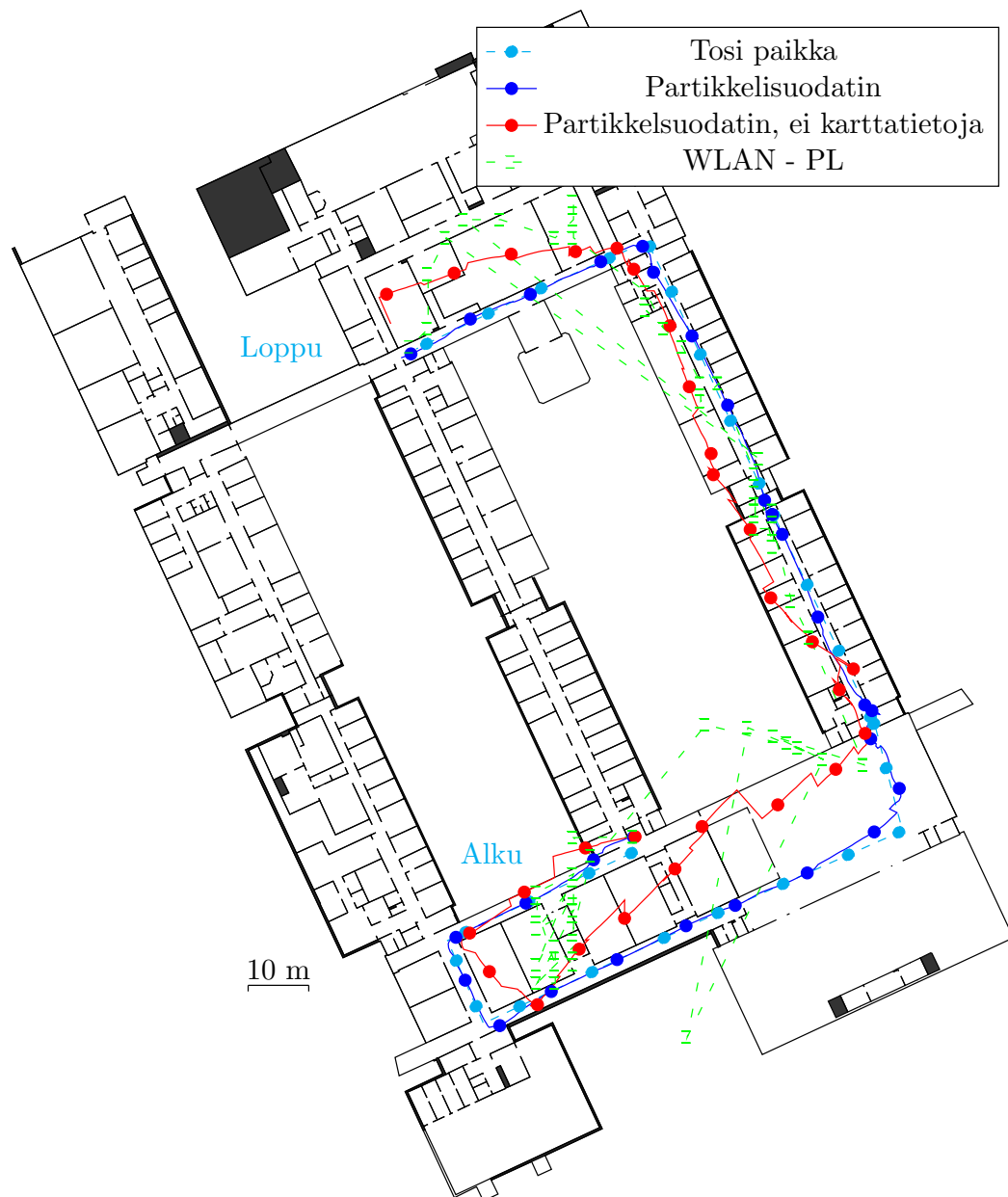
- [1] Cesare Alippi ja Manuel Roveri. An adaptive CUSUM-based test for signal change detection. Kirjassa *Circuits and Systems, 2006. ISCAS 2006. Proceedings. 2006 IEEE International Symposium on*, toukokuu 2006.
- [2] Vlad Badea ja Rikard Eriksson. Indoor navigation with pseudolites (fake GPS sat.). Diplomityö, Linköpingin yliopisto, tammikuu 2005.
- [3] Stephane Beauregard. A helmet-mounted pedestrian dead reckoning system. *Applied Wearable Computing (IFAWC), 2006 3rd International Forum on*, ss. 1–11, maaliskuu 2006.
- [4] Abdelmoula Bekkali, Horacio Sanson ja Mitsuji Matsumoto. RFID Indoor Positioning based on Probabilistic RFID Map and Kalman Filtering. Kirjassa *Wireless and Mobile Computing, Networking and Communications*, s. 21, lokakuu 2007.
- [5] A. Bose ja Chuan Heng Foh. A practical path loss model for indoor wifi positioning enhancement. Kirjassa *Information, Communications Signal Processing, 2007 6th International Conference on*, ss. 1–5, joulukuu 2007.
- [6] Kenneth C. Cheung, Stephen S. Intille ja Kent Larson. An inexpensive Bluetooth-based indoor positioning hack. 2006.
- [7] Dan Crisan ja Arnaud Doucet. A survey of convergence results on particle filtering methods for practitioners. Kirjassa *IEEE Transactions on signal processing*, osa 50, maaliskuu 2002.
- [8] André Emilsson. Indoor navigation using an iPhone. Diplomityö, Linköpingin yliopisto, 2010.
- [9] F. Evennou, F. Marx ja E. Novakov. Map-aided indoor mobile positioning system using particle filter. Kirjassa *Wireless Communications and Networking Conference, 2005 IEEE*, osa 4, ss. 2490–2494, maaliskuu 2005.
- [10] S. Gansemer, S. Hakobyan, S. Puschel ja U. Grosmann. 3d wlan indoor positioning in multi-storey buildings. Kirjassa *Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, 2009. IDAACS 2009. IEEE International Workshop on*, ss. 669–672, syyskuu 2009.
- [11] Thomas Gigl, Gerard J.M. Janssen, Vedran Dizdarevic, Klaus Witrisal ja Zoubir Irahauten. Analysis of a UWB Indoor Positioning System Based on Received Signal Strength. Kirjassa *4th Workshop on Positioning, Navigation and Communication*, maaliskuu 2007.

- [12] Pierre-Yves Gilliéron ja Bertrand Merminod. Personal navigation system for indoor applications. Kirjassa *Proceedings of the 11th IAIN World Congress, Berlin, Germany*, lokakuu 2003.
- [13] Saurabb Godha, Gérald Lachapelle ja Elizabeth Cannon. Integrated GPS/INS system for pedestrian navigation in a signal degraded environment. *International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS) 2006*, syyskuu 2006.
- [14] N.J. Gordon, D.J. Salmond ja A.F.M. Smith. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *Radar and Signal Processing, IEE Proceedings F*, 140(2):107–113, huhtikuu 1993.
- [15] Ville Honkavirta, Tommi Perälä, Simo Ali-Löytty ja Robert Piché. A comparative survey of WLAN location fingerprinting methods. Kirjassa *Proceedings of the 6th Workshop on Positioning, Navigation and Communication 2009 (WPNC'09)*, ss. 243–251, maaliskuu 2009.
- [16] Ville Huttunen ja Robert Piché. A monocular camera gyroscope. *Gyroscopy and Navigation*, 3(2):124–131, 2012.
- [17] The MathWorks Inc. Resizing and reshaping matrices. <http://www.mathworks.se/help/matlab/math/resizing-and-resaping-matrices.html> [Luettu 7.11.2012].
- [18] P. Kemppi, T. Rautiainen, V. Ranki, F. Belloni ja J. Pajunen. Hybrid positioning system combining angle-based localization, pedestrian dead reckoning and map filtering. Kirjassa *Indoor Positioning and Indoor Navigation (IPIN), 2010 International Conference on*, ss. 1–7, syyskuu 2010.
- [19] Laura Koski, Tommi Perälä ja Robert Piché. Indoor positioning using WLAN coverage area estimates. Kirjassa *2010 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, Zurich, Switzerland, syyskuu 2010.
- [20] Laura Koski, Robert Piché, Ville Kaseva, Simo Ali-Löytty ja Marko Hännikäinen. Positioning with coverage area estimates generated from location fingerprints. Kirjassa *Proceedings of the 7th Workshop on Positioning, Navigation and Communication 2010 (WPNC'10)*, ss. 99–106, Dresden Germany, maaliskuu 2010.
- [21] Nisarg Kothari, Balajee Kannan ja M Bernardine Dias. Robust indoor localization on a commercial smart-phone. Tekninen raportti CMU-RI-TR-11-27, Robotics Institute, Pittsburgh, PA, August 2011.
- [22] IndoorAtlas Ltd. Indoor atlas. <http://www.indooratlas.com/> [Luettu 13.12.2012].

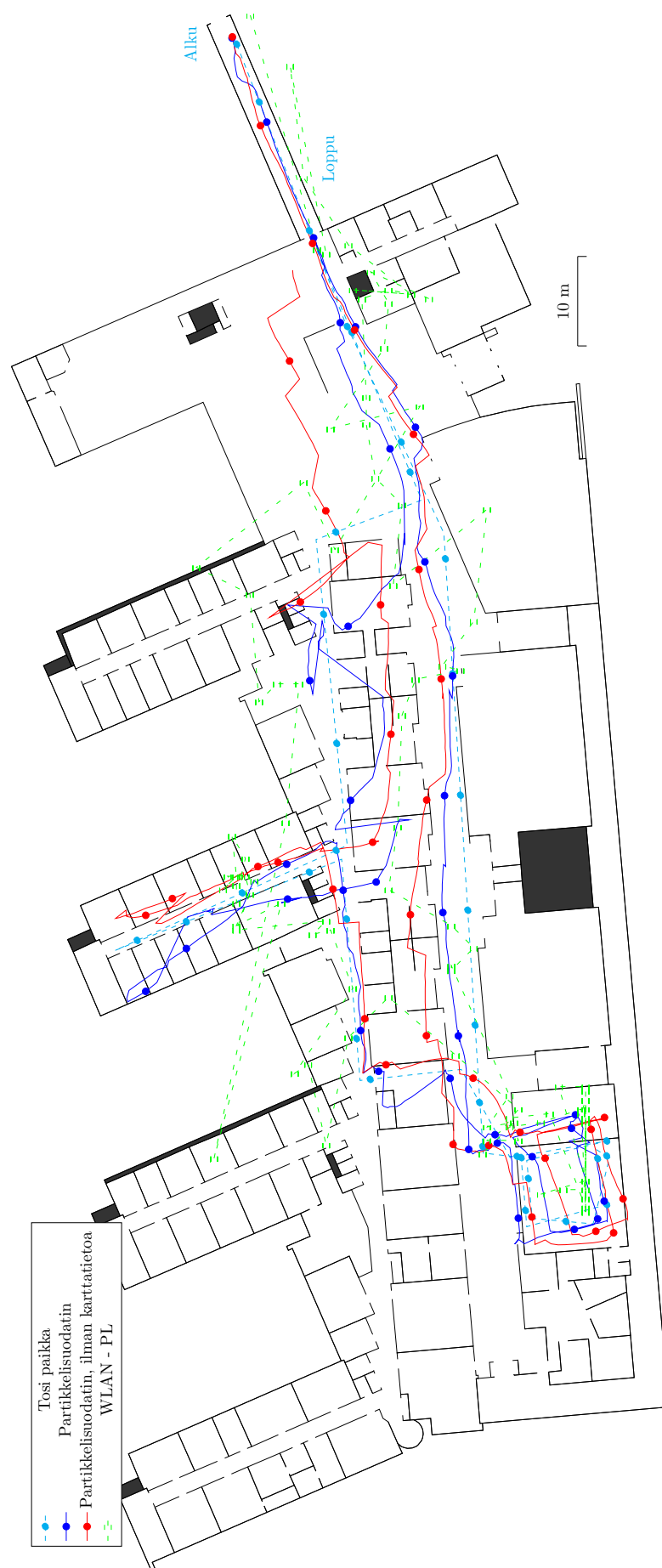
- [23] F. Marchal, J. Hackney ja K. W. Axhausen. Efficient map matching of large global positioning system data sets: Tests on speed-monitoring experiment in Zürich. *Transportation Research Record: Journal of the Transportation Research Board*, 1935(1):93–100, 1 2005.
- [24] Oleg Mezentsev, Jussi Collin ja Gérald Lachapelle. Pedestrian dead reckoning—a solution to navigation in GPS signal degraded areas? *Geomatica*, 59(2):175–182, 2005.
- [25] A.I. Mourikis ja S.I. Roumeliotis. Analysis of positioning uncertainty in simultaneous localization and mapping (slam). Kirjassa *Intelligent Robots and Systems, 2004. (IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, osa 1, ss. 13–20, syyskuu 2004.
- [26] Nokia. Accurate mobile indoor positioning industry alliance, called in-location, to promote deployment of location-based indoor services and solutions. lehdistöietote, elokuu 2012. <http://press.nokia.com/2012/08/23/accurate-mobile-indoor-positioning-industry-alliance-called-in-location-to-promote-deployment-of-location-based-indoor-services-and-solutions/>.
- [27] Henri Nurminen, Jukka Talvitie, Simo Ali-Löytty, Philipp Müller, Elena-Simona Lohan, Robert Piché ja Markku Renfors. Statistical path loss parameter estimation and positioning using RSS measurements. Kirjassa *Ubiquitous Positioning Indoor Navigation and Location Based Service (UPINLBS2012)*, lokakuu 2012.
- [28] Busra Ozdenizci, Kerem Ok, Vedat Coskun ja Mahmet N. Aydin. Development of an Indoor Navigation System Using NFC Technology. Kirjassa *Fourth International Conference on Information and Computing Science*, huhtikuu 2011.
- [29] Mohammed A. Quddus, Washington Y. Ochieng ja Robert B. Noland. Current map-matching algorithms for transport applications: State-of-the art and future research directions. *Transportation Research Part C: Emerging Technologies*, 15(5):312–328, 2007.
- [30] Matti Raitoharju, Toni Fadjukoff, Simo Ali-Löytty ja Robert Piché. Using unlocated fingerprints in generation of wlan maps for indoor positioning. Kirjassa *Proceedings of PLANS 2012 IEEE/ION Position Location and Navigation Symposium*, ss. 576–583, huhtikuu 2012.
- [31] Branko Ristic, Sanjeev Arulampalam ja Neil Gordon. *Beyond the Kalman Filter: Particle Filters for Tracking Applications*. Artech House Publishers, 2004.
- [32] B. Turgut ja R.P. Martin. Restarting particle filters: An approach to improve the performance of dynamic indoor localization. Kirjassa *Global Telecommunications Conference, 2009. GLOBECOM 2009. IEEE*, ss. 1–7, marraskuu 2009.

- [33] Greg Welch ja Gary Bishop. An introduction to the Kalman filter, 1995.
- [34] Martin Werner, Moritz Kessel ja Chadly Marouane. Indoor positioning using smartphone camera. Kirjassa *Indoor Positioning and Indoor Navigation (IPIN), 2011 International Conference on*, syyskuu 2011.
- [35] Widyawan, M. Klepal ja S. Beauregard. A backtracking particle filter for fusing building plans with pdr displacement estimates. Kirjassa *Positioning, Navigation and Communication, 2008. WPNC 2008. 5th Workshop on*, ss. 207–212, maaliskuuta 2008.
- [36] Lauri Wirola, Tommi A. Laine ja Jari Syrjärinne. Mass-market requirements for indoor positioning and indoor navigation. Kirjassa *2010 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, Zurich, Switzerland, syyskuu 2010.
- [37] Oliver Woodman ja Robert Harle. Pedestrian localisation for indoor environments. Kirjassa *Proceedings of the 10th international conference on Ubiquitous computing*, sarjassa *UbiComp '08*, UbiComp '08, ss. 114–123, New York, NY, USA, 2008. ACM.
- [38] XSens. XSens MTi. <http://www.xsens.com/en/general/mti> [Luettu 16.10.2012].

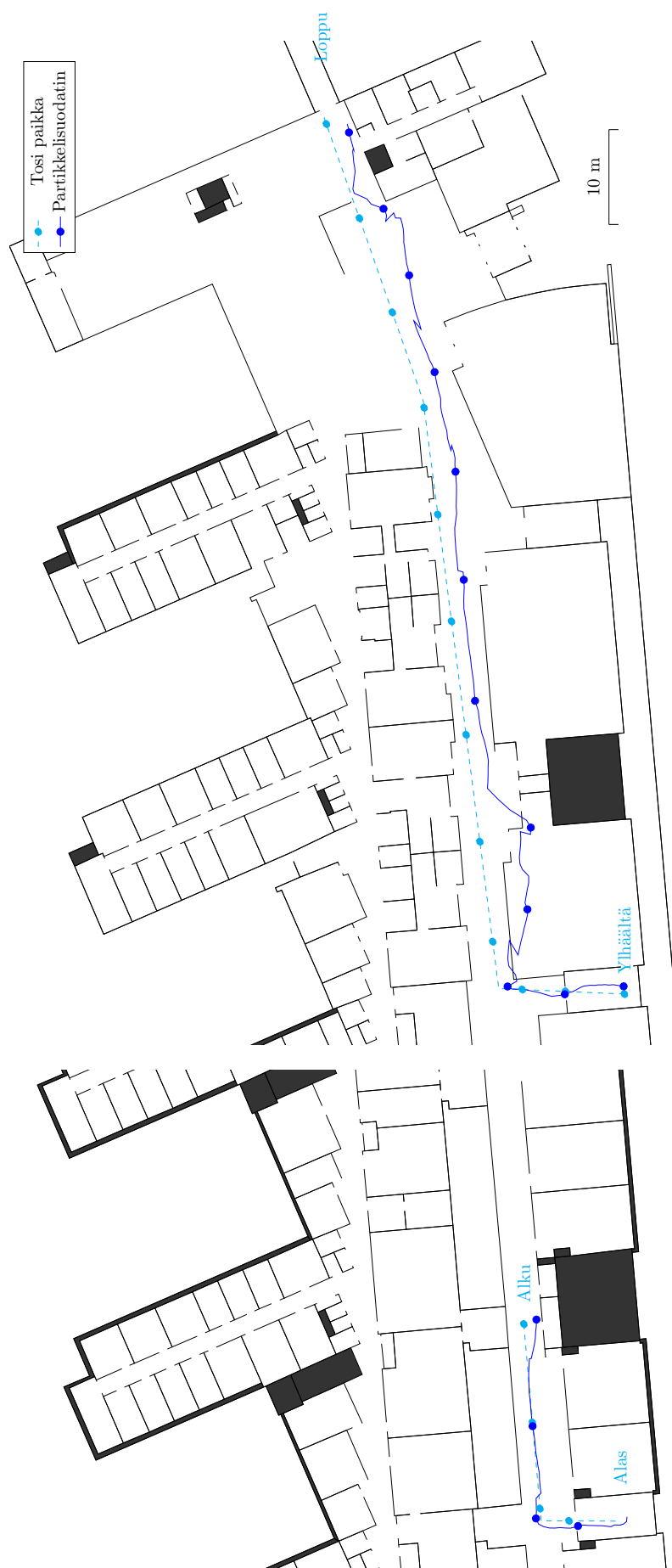
A. PAIKANNUSTULOKSIA



Kuva A.1: Kuva 4.15 suurempana



Kuva A.2: Kuva 4.13 suurempana



Kuva A.3: Kuva 5.2 suurempana